# HEALTH DATA MINING USING TENSOR FACTORIZATION: METHODS AND APPLICATIONS

A Dissertation Presented to
The Academic Faculty

By

Ioakeim Perros

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Computational Science & Engineering

Georgia Institute of Technology

August 2019

**HEALTH DATA MINING USING TENSOR FACTORIZATION: METHODS AND APPLICATIONS**

Approved by:

Dr. Jimeng Sun,
Advisor
School of Computational Science & Engineering
*Georgia Institute of Technology*

Dr. Richard Vuduc
School of Computational Science & Engineering
*Georgia Institute of Technology*

Dr. Haesun Park
School of Computational Science & Engineering
*Georgia Institute of Technology*

Dr. Walter F. Stewart
HINT Consulting

Dr. Evangelos E. Papalexakis
Department of Computer Science & Engineering
*University of California, Riverside*

Date Approved: April 29, 2019

## ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor, Jimeng Sun, for his invaluable guidance throughout this work. Jimeng has tirelessly shaped me into a researcher, being always there to offer timely and thoughtful advice across every single aspect of the Ph.D. journey. He has been persistent in teaching me how to conduct research, write technical papers and orally present research ideas. His enthusiasm and trust have been extremely inspiring. His commitment in constantly opening up opportunities for me to improve — be it through connecting with established researchers, providing with the opportunity to mentor fellow students and covering the travel costs of every single conference and workshop I attended — has been instrumental in my growth. During my last year at Georgia Tech, he has been extremely supportive with my job search, always there to provide with insightful feedback. Jimeng has also been a great friend, going above and beyond his role as an advisor and mentor. Working towards my Ph.D. under his guidance has been a great privilege.

I would also like to thank the rest of my committee members, Rich Vuduc, Haesun Park, Buzz Stewart and Vagelis Papalexakis, for providing with insightful feedback and comments regarding my thesis. In particular, Rich has been a role model throughout my Ph.D. and a great collaborator, mentor and friend. I am really grateful for all the time he has devoted in my growth and all the things he has taught me, ranging from high-performance computing concepts to writing technical papers and being a well-rounded academic citizen. Haesun taught me numerical linear algebra and further inspired me to work on factorization methods. Buzz has been an amazing mentor and friend and I feel extremely humbled and privileged to have been able to work with him. During my internship at Sutter Health and ever since, he has taught me so much about the field of health informatics and the process of identifying high-impact research problems. I would also like to particularly thank him for spending a lot of time to provide with detailed feedback on this document, which has given

me food for thought for years to come. Ever since my days at the Technical University of Crete in Greece, Vagelis has been an amazing mentor, colleague and friend. He is the one who first ignited my interest in matrix and tensor computations and has been an extremely knowledgeable and approachable mentor on tensor concepts and all things that matter. Our discussions and collaborations have been out of the most fruitful and enjoyable parts of this journey and he has been extremely supportive every step of the way.

I would like to acknowledge my collaborators for all their help and feedback: Ari Afshar, Anima Anandkumar, Casey Battaglino, Robert Chen, Dehua Cheng, Jee Choi, Christopher Defilippi, Bistra Dilkina, Joyce Ho, Furong Huang, J.B. Jones, Elias Khalil, Jiajia Li, Yan Liu, Uma Naresh Niranjan, Jyoti Pathak, Richard Peng, Bess Searles, Vaidy Sunderam, Michael Thompson, Peter Walker, Fei Wang, Li Xiong, Xiaowei "Sherry" Yan, Ping Zhang. I would also like to thank Tammy Kolda for all the guidance she has kindly been offering since the earliest stages of my Ph.D.; her precious feedback has had a significant impact on my research methodology. I also want to thank Minos Garofalakis, my Diploma and M.Sc. thesis advisor at the Technical University of Crete; it was Minos who brought me into the research world and enabled my first academic steps. Without his trust and support, none of my accomplishments so far would be possible.

I also want to thank all my fellow teammates at SunLab for all their feedback, help and fun moments we had: Edward Choi, Taha Bahadori, Mohammed Khalilia, Robert Chen, Kunal Malhotra, Sungtae An, Siddarth Biswal, Yanbo Xu, Jeff Valdez, Ari Afshar, Rahul Duggal, James Mullenbach, Olivier Deiss, Yu Jing, David Kartchner, Tianfan Fu, Hang Su, Zhaoming Wu, Yuyu Zhang.

I would also like to thank the personal friends I made in Atlanta: Yannis Dialynas, John Kimionis, Filippos Tagklis and Theodore Panagiotopoulos, for always being there for me and making this journey so enjoyable. I am also grateful to Ioannis Demertzis and Sofia Nikolakaki for being so supportive in all of my pursuits.

Finally, I would like to thank my parents, Anastasia and Manoussos, and my brother,

Iosif, for their unconditional love and sacrifices they have made in order to enable me to follow my dreams. Last, but certainly not least, I want to thank the love of my life, Athina. You have always been there for me, despite the thousands of miles between us. Your support, love and thoughtfulness mean the world to me.

## TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

## Summary

The increasing volume and availability of healthcare and biomedical data is opening up new opportunities for the use of computational methods to improve health. However, the data are diverse, multidimensional and sparse, posing challenges to the extraction of clinically-meaningful relations and interactions. For example, the electronic health records (EHRs) of patients contain time-stamped occurrences of diverse features (e.g., diagnoses, medications, procedures) as well as information about relationships among different types of features (e.g., identifying the subset of medications prescribed to treat a certain diagnosis). Such EHR data can be utilized to identify patient cohorts sharing common conditions without expert supervision, a task known as unsupervised phenotyping. Tensors, which are generalizations of matrices for higher orders, can naturally express the multidimensional data relationships inherent in the EHR. Tensor factorization encompasses a set of tools which can capture the latent correlation structure among diverse feature sets. For example, in the context of phenotyping, tensor factorization can be utilized to identify clinically-meaningful patient groups, along with succinct feature profiles distinguishing one group of patients from another.

In this dissertation, we expose how tensor factorization can be leveraged to tackle several important problems in healthcare and biomedicine. We also identify multiple significant methodological challenges in fully harnessing the capacity of tensor factorization for the problems at hand and develop algorithms to tackle them. In particular, we focus on the following problems:

- Drug-perturbed, tissue-specific gene expression prediction, where we demonstrate how tensor factorization can be used to model the interactions between drugs, genes and tissues in an efficient manner.

- Unsupervised phenotyping through EHRs, in the context of which we advance existing tensor factorization methods so that: a) they are fast and scalable to use for large patient

cohorts of hundreds of thousands of patients; and b) they yield interpretable output, easy to be communicated to a clinical expert.

- Automating understanding of physician desktop work. Therein, we demonstrate how tensor factorization can be used to substantially compress audit EHR logs, offering an intuitive categorization of user actions that can be used for workflow analysis.

# CHAPTER 1

# INTRODUCTION


**Summary**

Through this dissertation, we expose how tensors (i.e., generalizations of matrices for higher orders) and tensor factorization (a set of methods capturing and organizing the latent correlation structure in multidimensional data) can be applied and extended to tackle several important problems in healthcare and biomedicine.


The first of the problems we consider is to accurately predict drug-perturbed and tissue-specific gene expression values which reflect how drugs, genes and tissues interact with one another. The expression values are indicative of whether the corresponding drug could potentially treat a tissue-specific disease [1]. An accurate prediction of those expression values can be impactful for drug repurposing, by computationally identifying that an already-approved drug may have previously-unknown therapeutic capabilities [2]. Then, focusing on challenges in the healthcare domain, we tackle the EHR-based unsupervised phenotyping problem [3]. This problem refers to grouping patients sharing a similar condition without expert supervision. Efficient unsupervised phenotyping can be impactful for several downstream applications [4, 5]; one of them is clinical decision support, where physicians would be able to quickly identify similar patients and make more informed decisions, when a patient does not meet textbook treatment guidelines [4, 6, 7]. In the context of unsupervised phenotyping, we improve integer-constrained tensor factorization so that it can scale to hundreds of thousands of data samples, thus enabling the extraction of succinct and interpretable feature summaries (i.e., phenotype definitions) along with the associated patient subgroups from large cohorts. In order to extract insights which may lead to a better understanding of disease progression [8, 9], we further applied and extended tensor factorization

to extract not only phenotype candidates (i.e., patient groups and associated feature summaries), but also information about the timing and sequencing of those phenotypes over time (a task we call as temporal phenotyping [10]). We also expose how such tensor-based temporal phenotyping can be: a) applied to extract phenotypes of medically-complex children; b) scaled up to hundreds of thousands of patients by exploiting the sparse nature of EHR data; and c) extended to extract interpretable integer-based feature summaries indicating the phenotype definitions as well as real-valued temporal trends of phenotypes. Finally, we tackle the problem of automating the understanding of physician desktop work. In this context, we demonstrate how the EHR activity log file could be processed through tensor factorization to derive an intuitive categorization of EHR user actions that can be used for workflow analysis and improvement.

The data elements utilized in all the problems tackled as part of this thesis share several important properties: a) they are diverse in the sense of being described by multiple different sets of features (e.g., EHR data contain patient, diagnosis, medication, procedure and other information); b) they are multidimensional, due to the existence of high-order interconnections among different sets of features (e.g., a subset of medications is prescribed to treat a certain diagnosis for a given patient); and c) they are sparse (e.g., very few medical features occur per patient encounter as compared to the total number of features).

Tensor factorization [11, 12, 13, 14] is a set of methods which has been successfully used in diverse applications, such as social network analysis [15], text mining [16], image processing [17], recommendation systems [18], brain data analysis [19, 20] and healthcare analytics [21, 22], in order to extract the latent correlation structure in diverse, multidimensional and oftentimes sparse input data. A common paradigm among tensor factorization methods is the distillation of an input tensor to intuitive, low-dimensional representations. For example, we can derive a patient $\times$ diagnosis $\times$ medication tensor through EHR data, where the $(i, j, k)$ cell reflects the number of times the $i$-th patient has been prescribed the $k$-th medication to treat the $j$-th diagnosis. A popular tensor factorization model (CP [23])

would distill this patient $\times$ diagnosis $\times$ medication tensor into low-dimensional representations which may be interpreted as: a) soft phenotype assignments, where patients can have multiple phenotype assignments with different degrees of association; b) diagnosis and medication-based definitions of candidate phenotypes in terms of ensembles of correlated input features [24].

Tensor factorization belongs to the family of linear dimensionality reduction methods [25]. The advantage of those approaches over non-linear ones (e.g., autoencoders [26]) lies in that the output factors are linear transformations of the original input features, thus enhancing the ability to interpret what those factors signify w.r.t. the known set of input features (a property known as feature attribution [27]). On the other hand, simpler linear dimensionality reduction approaches such as matrix factorization (e.g., PCA) are not as suitable either when modeling interconnected high-order (order $> 2$) relations among different feature sets. Using matrix factorization for input data that can be most naturally represented as a tensor would require reshaping the tensor to a matrix which leads to mixing up the input features (e.g., a tensor of size $I \times J \times K$ would be reshaped to a $I \times J * K$ matrix). As such, the effect of a single input feature (e.g., a particular diagnosis) would be associated with multiple coordinates of an output factor matrix, which hinders interpretability. Overall, for the problems we tackle in this dissertation, we argue that tensor factorization strikes a proper balance between: a) model sophistication, in order to avoid unnatural transformations that would turn the factorization into a simpler problem; and b) model simplicity, in the sense that the linear relationship between the input and output factors promotes an intuitive model interpretation.

Below, we provide a more detailed introduction for each one of the problems tackled as part of this thesis by highlighting the motivation behind each one of them.

**Drug-perturbed, cell-specific gene expression prediction**

First, we are motivated by a challenge arising in the field of chemogenomics [1], that is

[1] Chemogenomics is defined as the study of the genomic and/or proteomic response of an intact biological system to chemical compounds and is central in drug discovery [28].

to identify new disease indications for approved drugs, also known as drug repositioning. Repurposing existing, approved drugs is expected to reduce the cost of developing new ones, which is estimated to be more than $1 billion [29] per drug. Repositioning can also speed up the drugs' adoption due to the fact that they have already undergone a rigorous safety testing required by the Food and Drug Administration [2].

Gene expression measurements can be used to measure the molecular activity of a drug in a biological system [30]. Differential gene expression measures the difference in gene expression between the drug-perturbed tissues and the controls. Differential expression values with large magnitude are indicative of significant change in the cells' behavior, meaning the drug could potentially treat the corresponding disease [1]. Despite the recent expansion of publicly available chemogenomic data providing such expression values over many drug-disease combinations [31], there still exist substantial amounts of missing expression values in the combinatorial space across drugs and diseases [32]. Such missing data exist mainly due to the high cost associated with producing expression values. To computationally address this challenge, we need accurate methods filling those gaps, i.e., predicting the expression values for (drug, gene, tissue) triads. Another notable challenge is provide accurate predictions for entirely new drugs, for which no measurement is available. To do so, we could exploit domain knowledge reflecting the similarity of new drugs with existing ones in the training set. In that way, instead of directly conducting many expensive lab experiments measuring the expression for a new drug, we could try to estimate its treatment effect first computationally. Then, we could focus on a small subset of targeted drug trials and cut down the corresponding costs. Through Polyadic Regression [33], we tackle the above challenges as described in Chapter 2. This work was presented at the 2017 SIAM International Conference on Data Mining (SDM).

**Scalable unsupervised phenotyping through electronic health records**

Next, we are focusing on the EHR-based unsupervised phenotyping problem, which is a central topic of this dissertation. Phenotyping is defined as grouping patients together that

have a similar condition whether this is defined within a specific disease (e.g., prediagnostic heart failure) or for a group of related diseases. The notion of searching for phenotypes is consistent with the recognition that people with the same named disease can differ substantially by the underlying pathophysiology. Phenotyping based on EHRs refers to identifying patients with specific, clinically-meaningful characteristics from large volumes of imperfect practice-based data [3]. The ability to quickly identify patients with particular traits by using common definitions is fundamental for a variety of target tasks. We list a few notable examples: clinical decision support, where a physician could instantaneously identify patients sharing common traits and interventions, enabling her to make more informed decisions, in the absence of gold-standard evidence [34, 4]; automating epidemic surveillance by estimating rates of a particular infection (e.g., the Centers for Disease Control and prevention need to automatically and quickly find patients with particular traits across geographies) [35]; genetic research can be empowered by large sample sizes of well-defined phenotypes, in order to identify genetic variations between cases and controls [36, 37]; phenotyping has the potential to streamline clinical trial recruitment as it may be used to determine patient eligibility [5] (e.g., by pharmaceutical companies).

Manual patient phenotyping is impractical for large cohorts, since it requires labor-intensive, time-consuming chart reviewing, which needs periodic revision [38]. Indicatively, the 2008 revision of the World Health Organization (WHO) classification of lymphoid neoplasms lasted more than a year, involving an eight-member steering committee and over $130$ pathologists and hematologists [39]. Also, only $1400$ cases were reviewed for this revision, which may result in substantial selection biases [22]. As such, automated, scalable phenotyping approaches are fundamental in order to both remove substantial burden from the domain experts and eliminate selection biases.

Even if they are not originally designed for research use, we are motivated to exploit EHRs to fulfill an automated fulfillment of phenotyping due to the dramatic increase in the EHR adoption [36]. As an example, 84% of the US hospitals had adopted a basic EHR

system as of 2015 [40]. This adoption results in massive cohorts of both structured (e.g., diagnoses codes) and unstructured data, such as clinical notes, being readily available for research purposes. Exploiting observational EHR data for phenotyping is arguably a more cost-effective direction than setting up Randomized Control Trials (RCTs) to accumulate cohorts of similar sizes [41].

Despite the aforementioned potential of exploiting EHRs for research purposes, EHR-based phenotyping is still grounded on a *secondary analysis* of EHRs, which have been originally designed for billing purposes [38]. As such, the purpose behind the design of some of the codes may be biased towards satisfying some financial incentive [42], rather than precisely describing a certain disease. An example is the existence of "unspecified" codes (e.g., 041.9 stands for unspecified bacterial infections) in the International Classification of Diseases [43], which is the most widespread approach of diagnosis coding in the EHR. Another issue is that although the timing of a patient visit is an important and meaningful feature (e.g., all diseases change over time), timing is an intricate feature to model because all clinical visits are essentially unscheduled; unscheduled means that there is no structure to the timing of clinical events other than that one follows another [44].

As a result of the above and other issues, EHR-based phenotyping is far more challenging than a simple code search and sophisticated methods which can distill heterogeneous data into clinically-interpretable descriptions are needed [7]. Phenotyping based on expert-defined rules is widely adopted, due to the simplicity of the resulting rules, which are often expressed in form of a decision tree and boolean logic. Still, the effort and time for developing such algorithms is significant, requiring both clinical and informatics knowledge [45]. Also, expert-defined rules cannot identify novel phenotypes, which are not first envisioned by a researcher; as such, they have limited scope for exploratory analysis [46]. Similar issues are faced by supervised phenotyping approaches, where patient cases need to be manually labeled through chart review [45, 47].

For the above reasons, we are motivated to study the unsupervised EHR-based pheno-

typing problem, which tremendously reduces the time needed for manual chart review [45]; it also needs close clinical expert involvement in the phenotypes' validation, as no clear ground truth is given [7].

In Chapter 3, we focus on the unsupervised EHR-based phenotyping problem, where a fixed observation window is considered for each patient (e.g., defined based on clinical knowledge). For this window, we are interested in extracting succinct phenotype definitions and associated patient clusters out of EHR data, which will be explainable and useful to medical experts. A common way of forming input data for unsupervised phenotyping is to produce co-occurrence count tensors. For example, a patient $\times$ diagnoses $\times$ medication tensor can be formed [24], where each $(i, j, k)$ cell reflects the number of times the $i$-th patient has been prescribed the $k$-th medication to treat the $j$-th diagnosis. In [48], we observe that factorizing such a tensor into real-valued factors as done by prior work (e.g., [24, 21, 49, 50, 51]) makes it hard to assess the relative importance or frequency between different factor elements (e.g., different medical features). For example, it is hard to understand how much more frequent is a certain diagnostic code over another within the same phenotype. This is mostly due to the arbitrary possible ranges and relative differences between real-valued elements. To tackle this interpretability issue in a scalable manner, and inspired by clinical scoring systems [2], we propose SUSTain [48], an integer tensor factorization framework for integer input tensors. The factor matrices are constrained to a small integer set; due to the integer nature of input data, the factor scores can be interpreted as distinct frequency levels. Extracting such frequency levels for medical features is crucial for phenotyping: a) increased frequency oftentimes hints increased severity of the corresponding condition due to the inherent bias in EHRs towards recording more severe patient conditions [52, 53, 54, 55, 56] and b) increased frequency boosts the confidence regarding the existence of the corresponding condition in the associated patients' records (e.g., in rule-based phenotyping, one of the inclusion criteria is oftentimes the occurrence

[2] In MDCalc, one can find a vast amount of such scoring systems used in medicine.

of at least 3 occurrences of a diagnostic code [57]). SUSTain [48] was presented at the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2018).

In Chapter 4, we tackle the challenge of modeling and utilizing the longitudinal nature of EHR data, in order to extract both the phenotype definitions and assignments to patients, as well as information regarding when a phenotype occurred; such insights may lead to a better understanding of disease progression [8, 9]. We call this task as temporal unsupervised phenotyping [10]. For this task, we observe that the PARAFAC2 model [58] is a great fit. PARAFAC2 can account for the folded structure of the input EHR data, where each patient is represented by a sequence of encounters, each of which can be represented by a set of features recorded. In particular, it enables extracting information about the timing and sequencing of phenotypes without requiring any ad-hoc collapsing or aggregation of encounters over time which would align the events across different patients but may also obfuscate the temporal ordering and structure of individual encounters. Another favorable property of PARAFAC2 is that it guarantees model uniqueness under mild assumptions [59]. The issue with non-unique models lies in that there may exist many arbitrarily rotated versions of a solution yielding the same approximation error and it is unclear to the practitioner which one of those solutions provides the true latent factors. An open challenge was that the standard algorithm fitting the PARAFAC2 model was essentially designed for dense input data [60] and no scalable algorithm existed that could exploit the sparse nature of EHR data towards scaling-up computations for large amount of patients (e.g., hundreds of thousands). Our proposed SPARTan [51] method, described in Chapter 4 addresses those challenges. This work was presented at the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2017).

In Chapter 5, we demonstrate how the PARAFAC2 factorization (as implemented in [51]) can be applied and extended to distill the EHR data of a cohort of patients facing complex clinical manifestations (medically-complex children from Children's Healthcare of

Atlanta) into clinically-meaningful phenotype definitions, their temporal trends over time and associated patient groups. We identify four phenotype definitions, validated by a clinical expert. We also demonstrate the utility of the extracted patient representations towards identifying groups of patients with significant survival variations. This work was published in the Journal of Biomedical Informatics [61].

In Chapter 6, we propose the integer-constrained PARAFAC2 factorization and utilize that to efficiently identify interpretable pre-diagnostic subtypes of heart failure (HF). Precisely identifying the various stages and progress of pre-diagnostic HF is important due to the prevalence of the disease, its heterogeneity and the potential of developing adaptive treatment strategies based on a better understanding of pre-diagnostic heart failure patient progress. Through our proposed approach, we extract succinct integer-based phenotype definitions and patient assignments to phenotypes, as well as real-valued factors indicating the timing and sequencing of phenotypes. We quantitatively demonstrate the increased interpretability of the resulting phenotypes based on integer constraints, as compared to the corresponding real-valued model; to do so, we evaluate both in terms of model conciseness and feature overlap across different phenotypes.

**An automated way of understanding physician desktop work through EHR audit data** Finally, in Chapter 7, we explored how the EHR activity log file, a rich source of complex time-stamped data on desktop activities, could be processed to derive scalable and quantifiable physician level workflow measures. While substantial advances have been made to improving the quality of care through the effective use of electronic health records (EHR) data (e.g., [5]) a parallel advance has not evolved for use of log file data to understand variability in how desktop work is done and the time required to do this work. Log files, however, are voluminous and difficult to decipher. We explored how tensor factorization and in particular PARAFAC2 [58] as implemented in [51] could be used to substantially compress a log file so that it offers more intuitive and accessible data that could be used for workflow analysis. To this end, we identified activity log record clusters that represent

common physician tasks: 1) medications access, 2) notes access, 3) order entry access, and 4) diagnosis modification.

We also sought to determine if the same method could be used to identify variation in how physicians completed tasks without the need of ground-truth label information. To validate our approach, we selected documentation of notes as our focus, given that it is well established that two dominant methods are used for this purpose when using Epic EHR. That is, the means by which notes are accessed and documented is known. Our results reveal two distinct clusters of physicians accessing notes by either using the Visit Navigator or the Wrap-Up option, representing known workflows for this task. This work was presented at the AMIA 2018 Informatics Summit.

# CHAPTER 2

# POLYADIC REGRESSION AND ITS APPLICATION TO CHEMOGENOMICS


Summary

In this Chapter, we study a problem in the field of chemical genomics: in order to identify new disease targets for already-approved drugs (aka drug repositioning), we need to estimate the drugs' effectiveness on several tissue-specific diseases, as this is measured over various genes. We introduce Polyadic Regression, a method exploiting tensor factorization to efficiently learn the interactions across different drug, gene and tissue combinations. We demonstrate superior accuracy in predicting drug-perturbed, cell-specific gene expression measurements over prior work modeling high-order interactions among diverse feature sets.


We study the problem of Polyadic Prediction, where the input consists of an ordered tuple of objects, and the goal is to predict a measurement associated with them. Many tasks can be naturally framed as Polyadic Prediction problems. In drug discovery, for instance, it is important to estimate the treatment effect of a drug on various tissue-specific diseases, as it is expressed over the available genes. Thus, we essentially predict the expression value measurements for several (drug, gene, tissue) triads. To tackle Polyadic Prediction problems, we propose a general framework, called Polyadic Regression, predicting measurements associated with multiple objects. Our framework enables predictions in both of the following important use cases: a) predicting missing measurements (e.g., expression values for drug $x_1$, gene $y$ and tissue $z$), and b) predicting measurements for out-of-sample objects (e.g., expression values involving a drug $x_2$ which is not involved in any of the training instances). Our model is expressive, exploring high-order, polyadic interactions in an efficient manner. An alternating Proximal Gradient Descent procedure is proposed to

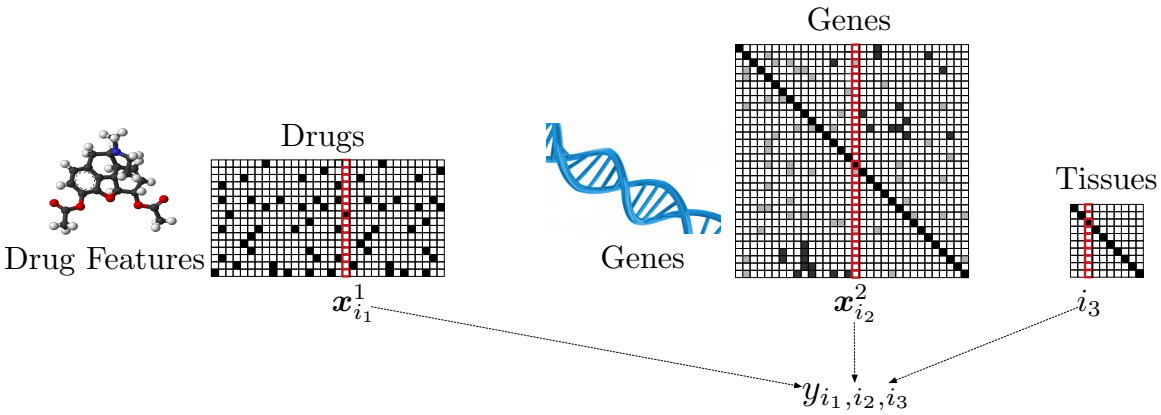



Figure 2.1: Example use case of Polyadic Prediction. We predict a measurement (expression value) associated with drug $i_1$, gene $i_2$ and tissue $i_3$, indicating whether the drug $i_1$ is effective in treating tissue $i_3$, w.r.t. gene $i_2$. Each drug is described by features capturing its chemical structure, and each gene is described by features reflecting its similarity with other genes.

fit our model. We perform an extensive evaluation using real-world chemogenomics data, where we illustrate the superior performance of Polyadic Regression over the prior art. Our method achieves an increase of $0.06$ and $0.1$ in Spearman correlation between the predicted and the actual measurement vectors, for predicting missing polyadic data and predicting polyadic data for new drugs, respectively.

## 2.1 Introduction

Dyadic data are measurements on dyads, i.e., ordered pairs, where each element of a pair originates from a finite set (i.e., data domain) of objects [62]. Typically, those data are represented in a matrix, where a measurement for a pair $(i, j)$ represents some form of relationship between the objects $i$ and $j$ (e.g., user $i$ and movie $j$ are two objects of user and movie data domains, and the corresponding rating reflects a certain relationship between the two objects). We generalize the notion of dyadic data to describe measurements associated with multiple objects (not only pairs), and call the corresponding data *polyadic*.

We study the problem of *Polyadic Prediction*, that is, predicting polyadic data. This problem covers many important use cases. For instance, in drug discovery, we are interested in the treatment efficacy of drugs on various types of tissue-specific diseases (e.g., different

types of cancer), as this is measured by the expression regarding all the available genes. While thousands of gene expression profiles (i.e., expression values for accessible genes) are available, substantial gaps remain in the combinatorial space across drugs and tissues. The reason for that is mainly the cost associated with producing expression values. An accurate prediction of missing expression values can empower a better understanding of drug mechanisms, a more precise identification of drug targets (i.e., specific proteins), as well as finding new uses for existing drugs (known as drug repositioning) [63].

We can naturally frame the challenge above as a Polyadic Prediction problem: our data domains correspond to the sets of objects: drugs, genes, and tissues. Given the observed measurements for triads involving a specific object from each data domain, we are interested in predicting unseen triadic data, i.e., unseen measurements for (drug $x_1$, gene $y$, tissue $z$) triads.

Another notable challenge is to provide accurate predictions for *entirely new objects* from some data domains, which are unseen during the training phase (cold-start problem [64]). For example, we may not have any available triads involving a certain drug $x_2$; still, it may be useful to estimate how this drug would interact with certain (gene, tissue) combinations. In particular, instead of directly conducting many expensive lab experiments measuring the expression for a new drug, we could try to estimate its treatment effect first computationally. Then, we could focus on a small subset of targeted drug trials and cut down the corresponding costs.

To enable predictions for new objects, we need some form of external knowledge for their data domain. We illustrate this use case in Figure 2.1, where we predict the measurement $y_{i_1,i_2,i_3}$ involving drug $i_1$, gene $i_2$ and tissue $i_3$. We assume that feature vectors are available for the drug data domain (e.g., $\boldsymbol{x}^1_{i_1}$ is the feature vector for drug $i_1$). Thus, we try to provide a prediction for $y_{i_1,i_2,i_3}$, even if the drug $i_1$ was not part of any observed measurement during training. In Figure 2.1, this holds accordingly for the genes' data domain.

Finally, in target applications involving polyadic data, a standard assumption is that significant interactions exist between objects across the data domains. For instance, we assume that the treatment effect of each drug is varying for different gene-tissue combinations. Regarding the example in Figure 2.1, this property is mathematically reflected through the interactions between drug with gene features and tissue objects. Thus, it is imperative to efficiently integrate those inter-domain interactions, which can be high-order for general polyadic data.

Dyadic Prediction has been studied in recent literature (e.g., [65, 66, 67], also mentioned as bilinear prediction or pair-input/pairwise learning). Still, those works limit their endeavors to dyadic data. Instead, we propose a new framework, which we call *Polyadic Regression*, to address the challenges introduced above. Our main contributions are as follows:

- We propose a **general** Polyadic Prediction framework, predicting measurements associated with multiple objects. The fitting algorithm adapts to various tasks (continuous/discrete) and constraints (e.g., sparse solutions via $\ell_1$ norm).
- Our framework is **inductive** [1]: it enables predictions for new objects which have not been encountered during training, tackling the so-called *cold-start problem*.
- Our model is **expressive** to fit the needs of Polyadic Prediction: it explores all the high-order interactions across different data domains. Its factorized version is designed to reduce its complexity and prevent overfitting.

We apply the proposed framework to problems in chemogenomics (i.e., chemical genomics). This field is facing many challenges that can be naturally framed as Polyadic Prediction problems, such as drug-induced, cell-specific gene expression prediction and drug-protein interaction prediction of various types [2]. Thus, we conduct an extensive evaluation on both synthetic and real, publicly-available chemogenomics data. The syn-

[1] Our distinction between the inductive and transductive settings follows the one provided in [66, 67].

thetic data experiment establishes the recovery of relevant features, in the presence of interactions. The real data case study on drug-induced, cell-specific gene expression prediction showcases the superior accuracy of Polyadic Regression as compared to the prior art in both of our target use cases. In particular, our approach improves the correlation (standard metric in gene expression analysis [68]) between the predicted and the actual measurement vectors by $0.06$ and $0.1$, in estimating missing polyadic data and predicting polyadic data involving new drugs, respectively.

## 2.2 Background

*Tensors* are high-order generalizations of matrices. The *order* of a tensor denotes the number of its modes (e.g., matrices are 2-order tensors). A *fiber* is a vector extracted from a tensor by fixing all modes but one. Considering a $d$-order tensor $\mathcal{S} \in \mathbb{R}^I$, where $I := I_1 \times \cdots \times I_d$, the index set of each individual mode $\mu$ is $I_\mu := \{1, \ldots, n_\mu\}, \mu \in \{1, \ldots, d\}$. *Matricization*, also called *reshaping* or *unfolding*, logically reorganizes tensors into other forms without changing the values themselves. The index set without mode-$\mu$ is $I^{(\mu)} := I_1 \times \cdots \times I_{\mu-1} \times I_{\mu+1} \times \cdots \times I_d$. Then, the *$\mu$-mode matricization* is a mapping from a tensor to a matrix: $\boldsymbol{S}^{(\mu)} : \mathbb{R}^I \rightarrow \mathbb{R}^{I_\mu \times I^{(\mu)}}$. As a result, the mode-$\mu$ fibers of the tensor become columns of a matrix. Given $\boldsymbol{U}_\mu \in \mathbb{R}^{J_\mu \times I_\mu}$, the *$\mu$-mode multiplication* (or $\mu$-mode product) is defined by $\boldsymbol{S} \times_\mu \boldsymbol{U}_\mu$ and its matricized version is $\boldsymbol{U}_\mu \boldsymbol{S}^{(\mu)} \in \mathbb{R}^{J_\mu \times I^{(\mu)}}$. Given matrices $\boldsymbol{U}_v \in \mathbb{R}^{J_v \times I_v}, v = 1, \ldots, d$, we can generalize this operation for all the tensor modes via the *multi-linear multiplication*, denoted as: $\mathcal{S} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \cdots \times_d \boldsymbol{U}_d \in \mathbb{R}^{J_1 \times \cdots \times J_d}$ [11].

## 2.3 Problem & Model Formulation

First, we formally introduce the Polyadic Prediction problem and our proposed model in its most general form. Next, we extend this model to cases when prior knowledge for certain data domains is not available. Then, we present a more efficient, factorized version of it which can easily handle high-order input ($\geq$ 3-order).

### 2.3.1 Problem Definition

Suppose our dataset consists of $N$ samples of the following form: $\{(i_1, i_2, \ldots, i_K), y_{i_1,i_2,\ldots,i_K}\}$, where $y_{i_1,i_2,\ldots,i_K}$ is an observed measurement involving the $K$ objects $i_1, i_2, \ldots, i_K$. We further assume that $i_k \in \{1, 2, \ldots, n_k\}, 1 \leq k \leq K$, that is, the object $i_k$ takes values from a finite set of objects with cardinality $n_k$. We call this set the $k$-th data domain, since it contains objects indexed in the $k$-th position of the ordered tuple $(i_1, i_2, \ldots, i_K)$. The goal of Polyadic Prediction is to learn a function to predict the value of unseen measurements $y_{i_1,i_2,\ldots,i_K}$, given the values of the observed ones.

First, in Section 2.3.2, we assume that external knowledge is available for all $K$ data domains. In that case, each object $i_k$ is described by a feature vector $\boldsymbol{x}^k_{i_k} \in \mathbb{R}^{d_k}$, where the superscript refers to the $k$-th data domain and $d_k$ is the size of its feature vectors. In Section 2.3.3, we relax this assumption.

### 2.3.2 Core Model

Our core regression model is

$$
\begin{aligned}
& f(\boldsymbol{x}^1_{i_1}, \boldsymbol{x}^2_{i_2}, \cdots, \boldsymbol{x}^K_{i_K}) \\
& = b + \underbrace{\sum_{k=1}^{K} (\boldsymbol{w}^k)^\top \boldsymbol{x}^k_{i_k}}_{\text{linear terms}} + \underbrace{\sum_{uv} (\boldsymbol{x}^u_{i_u})^\top \mathbf{S}^{uv} \boldsymbol{x}^v_{i_v}}_{\text{dyadic interactions}} \\
& \quad + \underbrace{\sum_{uvr} \mathcal{S}^{uvr} \times_1 \boldsymbol{x}^u_{i_u} \times_2 \boldsymbol{x}^v_{i_v} \times_3 \boldsymbol{x}^r_{i_r} + \cdots}_{\text{triadic interactions}} \\
& \quad + \underbrace{\mathcal{S} \times_1 \boldsymbol{x}^1_{i_1} \times_2 \boldsymbol{x}^2_{i_2} \cdots \times_K \boldsymbol{x}^K_{i_K}}_{\text{general polyadic interactions}}
\end{aligned}
\tag{2.1}
$$

where $b$ is a scalar offset, $\boldsymbol{w}^k \in \mathbb{R}^{d_k}$ is the vector capturing the linear feature effects of each $k$-th data domain, $\mathbf{S}^{uv} \in \mathbb{R}^{d_u \times d_v}$ is the matrix capturing dyadic feature interaction effects across the $u, v$ data domains, $\mathcal{S} \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_K}$ is a tensor capturing the $K$-way feature interaction effects across all the data domains. Our core model is expressive enough to

capture all the dyadic, triadic, and in general polyadic interactions emerging across features of various data domains.

We assume the matrices/tensors capturing interaction effects are order-independent w.r.t. the data domains, i.e., $\mathbf{S}_{ij}^{uv} = \mathbf{S}_{ji}^{vu}$, $\mathcal{S}_{ijk}^{uvr} = \mathcal{S}_{ikj}^{urv} = \mathcal{S}_{jik}^{vur} = \mathcal{S}_{jki}^{vru} = \mathcal{S}_{kij}^{ruv} = \mathcal{S}_{kji}^{rvu}$ and so on for the higher-order terms. That is, we model the interactions of each group of data domains with a single matrix/tensor, ignoring the rest possible data domain permutations.

### 2.3.3 Partial Induction

The model in Equation (2.1) implies that we have external knowledge available for all the data domains. We call this setting *complete induction*. However, this setting may not hold in practice. For example, we may have drug and gene features available, but no external information describing each one of the tissues. This setting is henceforth referred as *partial induction*.

First, consider a dyadic data example. In complete induction, the measurement for a dyad $(i_1, i_2)$ is given by

$$f(\boldsymbol{x}_{i_1}^1, \boldsymbol{x}_{i_2}^2) = b + \left(\boldsymbol{w}^1\right)^\top \boldsymbol{x}_{i_1}^1 + \left(\boldsymbol{w}^2\right)^\top \boldsymbol{x}_{i_2}^2 + \left(\boldsymbol{x}_{i_1}^1\right)^\top \mathbf{S}^{12} \boldsymbol{x}_{i_2}^2$$

We now assume that external knowledge is only available for the 1st data domain. In this case, we can predict a measurement associated with any object belonging to the 1st data domain (either it is included in the training set or not) and any (encountered during training) object belonging to the 2nd data domain. One typical analogue for this case is multi-label learning [69], where we have features describing the objects of the 1st, and a set of labels as the objects of the 2nd data domain. We follow the idea proposed in [70] and assume there are shared and individual components in each label predictor. More concretely, the possibility of assigning the label $i_2$ to object $i_1$ can be defined as

$$f_{i_2}(\boldsymbol{x}_{i_1}^1) \quad = \quad b + (\boldsymbol{w}^1)^\top \boldsymbol{x}_{i_1}^1 + (\boldsymbol{x}_{i_1}^1)^\top \boldsymbol{s}_{i_2}^{12}$$

where $\boldsymbol{w}^1$ is shared across all label predictors, and $\boldsymbol{s}_{i_2}^{12}$ is distinct for each label predictor. If we define $\mathbf{S}^{12} = [\boldsymbol{s}_1^{12}, \boldsymbol{s}_2^{12}, \cdots, \boldsymbol{s}_{n_2}^{12}] \in \mathbb{R}^{d_1 \times n_2}$, then we can re-write $f_{i_2}(\boldsymbol{x}_{i_1}^1)$ as

$$f_{i_2}(\boldsymbol{x}_{i_1}^1) \;\; = \;\; b + \left(\boldsymbol{w}^1\right)^\top \boldsymbol{x}_{i_1}^1 + \left(\boldsymbol{x}_{i_1}^1\right)^\top \mathbf{S}^{12} \boldsymbol{e}_{i_2}^2$$

where $\boldsymbol{e}_{i_2}^2 \in \mathbb{R}^{n_2}$ is a vector with only the $i_2$-the element being 1, all other elements are zeros (one-hot encoding) and $n_2$ is the cardinality of the 2nd data domain (i.e., number of distinct labels).

We generalize the idea above for the Polyadic Regression model as follows. Let $\mathcal{K} = \{1, 2, \cdots, K\} = \mathcal{K}_U \bigcup \mathcal{K}_F$ be the set of data domains. $\mathcal{K}_U$ is the set of data domains without external knowledge, $\mathcal{K}_F$ is the set of data domains with features describing their corresponding objects. In order to adapt Equation 2.1 to the partial induction setting, we define $\bar{\boldsymbol{x}}_{i_k}^k$ as either the feature vector of object $i_k$ from the $k$-th data domain, when $k \in \mathcal{K}_F$, or the one-hot encoding of size $n_k$ for the $i_k$ object, when $k \in \mathcal{K}_U$, i.e.,

$$\bar{\boldsymbol{x}}_{i_k}^k = \begin{cases} \boldsymbol{x}_{i_k}^k \in \mathbb{R}^{d_k}, & \text{if } k \in \mathcal{K}_F \\ \boldsymbol{e}_{i_k}^k \in \mathbb{R}^{n_k}, & \text{if } k \in \mathcal{K}_U \end{cases}$$

We also use $\bar{\boldsymbol{w}}^k$ to either denote the linear feature effects of the $k$-th data domain when $k \in \mathcal{K}_F$, or a zero vector which eliminates those effects when features are not available (when $k \in \mathcal{K}_U$), i.e.,

$$\bar{\boldsymbol{w}}^k = \begin{cases} \boldsymbol{w}^k \in \mathbb{R}^{d_k}, & \text{if } k \in \mathcal{K}_F \\ \mathbf{0}^{n_k} \in \mathbb{R}^{n_k}, & \text{if } k \in \mathcal{K}_U \end{cases}$$

Then, the model takes the form

$$\begin{aligned} & f(\bar{\boldsymbol{x}}_{i_1}^1, \bar{\boldsymbol{x}}_{i_2}^2, \cdots, \bar{\boldsymbol{x}}_{i_K}^K) \\ & = b + \sum_{k=1}^{K} (\bar{\boldsymbol{w}}^k)^\top \bar{\boldsymbol{x}}_{i_k}^k + \sum_{uv} (\bar{\boldsymbol{x}}_{i_u}^u)^\top \bar{\mathbf{S}}^{uv} \bar{\boldsymbol{x}}_{i_v}^v \\ & \quad + \sum_{uvr} \bar{\mathcal{S}}^{uvr} \times_1 \bar{\boldsymbol{x}}_{i_u}^u \times_2 \bar{\boldsymbol{x}}_{i_v}^v \times_3 \bar{\boldsymbol{x}}_{i_r}^r + \cdots \\ & \quad + \bar{\mathcal{S}} \times_1 \bar{\boldsymbol{x}}_{i_1}^1 \times_2 \bar{\boldsymbol{x}}_{i_2}^2 \cdots \times_K \bar{\boldsymbol{x}}_{i_K}^K \end{aligned} \tag{2.2}$$

If we define $\bar{d}_k$ as

$$\bar{d}_k = \begin{cases} d_k, & \text{if } k \in \mathcal{K}_F \\ n_k, & \text{if } k \in \mathcal{K}_U \end{cases}$$

then $\bar{\mathbf{S}}^{uv} \in \mathbb{R}^{\bar{d}_u \times \bar{d}_v}$, $\bar{\mathcal{S}}^{uvr} \in \mathbb{R}^{\bar{d}_u \times \bar{d}_v \times \bar{d}_r}$, $\bar{\mathcal{S}} \in \mathbb{R}^{\prod_k \bar{d}_k}$.

### 2.3.4 Factorizing Polyadic Interactions

Below, we first assume a complete induction setting, and then extend the discussion for the partial induction one. The core model presented in Equation 2.1 enjoys high expressive power in capturing all high-order interactions across different data domains. However, this approach suffers from high model complexity ($\mathcal{O}(d^K)$ for $K$ data domains and $d$ features per domain), which apart from efficiency issues, may result in a poor generalization performance.

As the pioneering work on Factorization Machines [71] suggested, it is reasonable to assume that feature interactions are not independent, and patterns are emerging among them. Those patterns imply a low-rank structure of the corresponding matrix/tensor reflecting interactions. To incorporate such a structure, one could augment the objective with a rank-minimizing constraint (e.g., trace norm minimization [72]). Still, this approach faces a huge model size problem for high-order interactions and is not feasible for our model.

Instead, we explicitly replace the parameters capturing interaction effects with their low-rank counterparts, i.e., products of sets of low-rank matrices or tensors. Moreover, we take the idea of shared patterns among features [71] a step further: we consider that shared structure also exists among interactions of different orders, so that, for instance, the low-rank approximation of $\S^{uv}$ has common terms with the one of $\mathcal{S}^{uvr}$. We mathematically translate this assumption to having a basis matrix $\mathbf{F}^k \in \mathbb{R}^{d_k \times m}$ capturing the low-rank structure of features for each $k$-th data domain. In that case, we would approximate $\S^{uv}$ as

$$\mathbf{S}^{uv} \approx \mathbf{F}^u (\mathbf{F}^v)^\top = \sum_{j=1}^{m} \mathbf{F}^u_j (\mathbf{F}^v_j)^\top \tag{2.3}$$

where $\mathbf{F}_j^u$ is the $j$-th column of $\mathbf{F}^u$. The formulation above implies that each dyadic effect (entries of $\S^{uv}$) is a result from aggregating the interactions (outer products) among the $m$ feature groups defined by the columns of $\mathbf{F}^u, \mathbf{F}^v$. It is equivalent to the bilinear model proposed in [67].

However, a pure generalization of Equation (2.3) for our model is restrictive. It would imply that the interactions between feature groups share the same contribution (i.e., weighting factor) for various data domain pairs or different orders. For a triadic data example, the weight of the interaction between $\mathbf{F}_1^u$ and $\mathbf{F}_1^v$ towards $\S^{uv}$ is restrained to be the same as the one between $\mathbf{F}_1^u$ and $\mathbf{F}_1^r$ towards $\S^{ur}$. Another restrictive assumption is that the subspaces defined by all the basis matrices $\mathbf{F}^k$ share the same dimension ($m$).

To tackle the above issues, we incorporate a scalar weight for each feature group interaction, capturing its significance towards the specific pair, triplet or higher combination of feature groups considered. We also permit a different dimension for the subspace corresponding to each basis matrix. Thus, Relation 2.3 is transformed to

$$\mathbf{S}^{uv} \approx \mathbf{F}^u \mathbf{C}^{uv} (\mathbf{F}^v)^\top = \sum_{j=1}^{m_u} \sum_{k=1}^{m_v} \mathbf{C}_{jk}^{uv} \mathbf{F}_j^u (\mathbf{F}_k^v)^\top \tag{2.4}$$

so that each dyadic interaction matrix $\S^{uv}$ is approximated by a tri-factorization [16]. For higher-order terms, we adopt the multi-way analogue of tri-factorization, which is the Tucker decomposition [73]:

$$\begin{aligned} \mathcal{S}^{uvr} &\approx \mathcal{C}^{uvr} \times_u \mathbf{F}^u \times_v \mathbf{F}^v \times_r \mathbf{F}^r \\ &\vdots \\ \mathcal{S} &\approx \mathcal{C} \times_1 \mathbf{F}^1 \times_2 \mathbf{F}^2 \cdots \times_K \mathbf{F}^K \end{aligned}$$

where $\mathcal{C}^{uvr} \in \mathbb{R}^{m_u \times m_v \times m_r}, \mathcal{C} \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_K}$ are low-rank core tensors, and $\mathbf{F}^k \in \mathbb{R}^{d_k \times m_k}$ with $m_k \ll d_k$.

The discussion above implies a complete induction setting. Consider now that we have no external knowledge for the $u$-th data domain. In that case, we no longer need to learn a

low-rank representation of features $\mathbf{F}^u$. However, it is still useful to learn how the objects from the $u$-th data domain interact with features/objects from other domains. To model this behavior, we fix $\mathbf{F}^u$ to be an identity matrix $\mathbf{I}^u \in \mathbb{R}^{n_u \times n_u}$ and Relation (2.4) becomes

$$\bar{\S}^{uv} \approx \bar{\mathbf{C}}^{uv}(\mathbf{F}^v)^\top$$

where $\bar{\S}^{uv} \in \mathbb{R}^{n_u \times d_v}$, $\bar{\mathbf{C}}^{uv} \in \mathbb{R}^{n_u \times m_v}$. This is trivially extended for higher-order terms.

## 2.4 Algorithm

### 2.4.1 Objective Formulation

For simplicity, we assume a complete induction setting and for notational convenience, we use $\{X_i, y_i\}_{i=1}^N$ to represent the $i$-th data sample, where $X_i = (\boldsymbol{x}_{i_1}^1, \boldsymbol{x}_{i_2}^2, \cdots, \boldsymbol{x}_{i_K}^K)$ and $y_i$ is the ground truth measurement of the objects involved in $X_i$. We define our objective function as the sum of a term minimizing the desired loss and another one intended to regularize the parameters. This objective can be summarized as

$$\mathcal{L} = \frac{1}{N}\sum_i \underbrace{\ell(f(X_i), y_i)}_{\mathcal{J}(\mathbf{u})} + \lambda \underbrace{\Omega(\{\boldsymbol{w}^k\}_{k=1}^K, \{\mathbf{S}^{uv}\}_{uv}, \cdots, \mathcal{S})}_{\mathcal{R}(\mathbf{u})} \tag{2.5}$$

where $\ell(f(X_i), y_i)$ is the regression loss of function $f$, $\Omega$ is an aggregation of the regularizations imposed on the model parameters and $N$ is the number of training samples.

### 2.4.2 Objective minimization

The most straightforward choice of a method minimizing the objective in Equation (2.5) would be that of an alternating Gradient Descent (GD). Such a choice would suffice in cases when the regularization function is smooth, such as the squared $\ell_2$-norm.

However, this choice falls short for *nonsmooth* regularizers [74], such as the $\ell_1$-norm which is the standard method to induce sparsity in the solution. In those cases, we have to handle the regularizer as a distinct entity, which can possibly be non-differentiable. To do

so, we can adopt proximal gradient methods [75] in each alternating iteration, which aim to solve optimization problems of the following form

$$\min_{\boldsymbol{u}\in\mathcal{H}} \mathcal{J}(\boldsymbol{u}) + \mathcal{R}(\boldsymbol{u}) \tag{2.6}$$

where $\mathcal{J}$ is convex and differentiable with Lipschitz continuous gradient, $\mathcal{R}$ is a convex, lower semi-continuous function which is possibly non-differentiable, and $\mathcal{H}$ is some set, typically a Hilbert space. The correspondence between the objectives (2.5) and (2.6) is clear: the loss function corresponds to $\mathcal{J}(\boldsymbol{u})$ and $\mathcal{R}(\boldsymbol{u})$ is considered as the function reflecting any regularizations.

In proximal methods, $\boldsymbol{u}$ minimizes $\mathcal{J}(\boldsymbol{u})+\mathcal{R}(\boldsymbol{u})$ if and only if $0 \in \partial_{\boldsymbol{u}}(\mathcal{J}(\boldsymbol{u})+\mathcal{R}(\boldsymbol{u}))$, where $\partial$ is the sub-differential operator. Given a convex function $\psi : \mathcal{H} \to \mathbb{R}$, we can define its proximal operator $\text{prox}_{\psi} : \mathcal{H} \to \mathcal{H}$ as $\text{prox}_{\psi}(\boldsymbol{z}) = \arg\min_{\boldsymbol{u}\in\mathcal{H}} \psi(\boldsymbol{u}) + \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{z}\|_2^2$, which can be seen as a generalization of a projection [76]. Then, the Proximal Gradient (PG) method [75] dictates the following update rule to solve the objective (2.6)

$$u^{k+1} := \text{prox}_{\gamma\mathcal{R}}\left(u^k - \gamma^k \nabla \mathcal{J}(u^k)\right) \tag{2.7}$$

where $k$ denotes the current iteration and $\gamma^k > 0$ is a step size, which can be found through line search. In the Supplementary Material [2], we include details regarding both the implementation and the various possible choices of losses and regularization functions.

## 2.5 Experimental Analysis

Due to space limitations, we include our synthetic data experiments in the Supplementary Material. In the following, we address a real-world challenge arising in the field of chemical genomics: the prediction of drug-induced and cell-specific gene expression values.

[2] www.cc.gatech.edu/~iperros3/pdf/sdm17-supp.pdf

### 2.5.1 Background

First, we briefly present some background related to our target application. Differential gene expression profiling of *in vitro* drug perturbations refers to the process of measuring the *difference* in gene expression of a certain cell culture (e.g., cells from brain affected with cancer) before and after treating it with a specific drug. A large differential expression value is indicative of a significant change in the cells' behavior, meaning that the drug could potentially treat the corresponding disease. The methodology described above (known as chemogenomic profiling) has provided powerful insights towards several important tasks, such as better understanding of drug mechanisms [77] and drug repurposing [30]. At the same time, we witness an expansion of the publicly available chemogenomic data through the Library of Integrated Cellular Signatures (LINCS) program [31]: they provide drug-induced and cell-specific gene expression measurements for roughly $1000$ genes, which are known to be maximally predictive of the expression of the rest of the available genes [31].

As we discussed in Section 2.1, the data mentioned above have inherently many missing values, since drugs are often measured only for a subset of tissues. Thus, a significant challenge is to estimate the missing expression values. Another important goal is to enable predictions for new drugs, which are unseen during training. The experiments illustrating the superiority of Polyadic Regression in terms of predicting missing values and expression values for new drugs are provided in Sections 2.5.4 and 2.5.5, respectively.

### 2.5.2 Formulation for drug effect prediction

Next, we present our formulation towards drug-induced, cell-specific gene expression prediction. Consider that we have $n_1$ drug objects, $n_2$ gene objects and $n_3$ tissue objects. We used external knowledge for the drug and gene data domains, but no knowledge is available for the tissues (partial induction on the domains of drugs and genes). Thus, $\boldsymbol{x}^1_{i_1} \in \mathbb{R}^{d_1}, \boldsymbol{x}^2_{i_2} \in \mathbb{R}^{d_2}$ are the feature vectors for drug $i_1$ and gene $i_2$ respectively. We also incorporate the low-rank assumptions presented in Section 2.3.4. Thus, the expression

value on gene $i_2$ perturbed by drug $i_1$ on tissue $i_3$ is calculated as

$$
\begin{aligned}
& f_{i_3}(\boldsymbol{x}^1_{i_1}, \boldsymbol{x}^2_{i_2}) = b + (\boldsymbol{w}^1)^\top \boldsymbol{x}^1_{i_1} + (\boldsymbol{w}^2)^\top \boldsymbol{x}^2_{i_2} \\
& + (\boldsymbol{x}^1_{i_1})^\top \mathbf{F}^1 \mathbf{C}^{12} (\mathbf{F}^2)^\top \boldsymbol{x}^2_{i_2} \\
& + (\boldsymbol{x}^1_{i_1})^\top \mathbf{F}^1 \bar{\mathbf{C}}^{13} \boldsymbol{e}^3_{i_3} + (\boldsymbol{x}^2_{i_2})^\top \mathbf{F}^2 \bar{\mathbf{C}}^{23} \boldsymbol{e}^3_{i_3} \\
& + (\bar{\mathcal{C}} \times_1 \mathbf{F}^1 \times_2 \mathbf{F}^2 \times_3 \mathbf{I}^3) \times_1 \boldsymbol{x}^1_{i_1} \times_2 \boldsymbol{x}^2_{i_2} \times_3 \boldsymbol{e}^3_{i_3}
\end{aligned} \tag{2.8}
$$

where $\mathbf{F}^1 \in \mathbb{R}^{d_1 \times m_1}$, $\mathbf{F}^2 \in \mathbb{R}^{d_2 \times m_2}$, $\mathbf{C}^{12} \in \mathbb{R}^{m_1 \times m_2}$, $\bar{\mathbf{C}}^{13} \in \mathbb{R}^{m_1 \times n_3}$, $\bar{\mathbf{C}}^{23} \in \mathbb{R}^{m_2 \times n_3}$, $\bar{\mathcal{C}} \in \mathbb{R}^{m_1 \times m_2 \times n_3}$, $\mathbf{I}^3 \in \mathbb{R}^{n_3 \times n_3}$ is an identity matrix and $\boldsymbol{e}^3_{i_3} \in \mathbb{R}^{n_3}$ is the one-hot encoding of tissue $i_3$. Since our measurements are continuous, we select the squared loss and solve for the model parameters by optimizing the following objective

$$
\begin{aligned}
\min_{\substack{\boldsymbol{w}^1, \boldsymbol{w}^2, \mathbf{F}^1, \mathbf{F}^2, \\ \mathbf{C}^{12}, \bar{\mathbf{C}}^{13}, \bar{\mathbf{C}}^{23}, \bar{\mathcal{C}}}} & \frac{1}{N} \sum_{(i_1, i_2, i_3) \in \mathcal{D}} \left( f_{i_3}(\boldsymbol{x}^1_{i_1}, \boldsymbol{x}^2_{i_2}) - y_{i_1, i_2, i_3} \right)^2 \\
& + \lambda_1 \big( \Omega(\boldsymbol{w}^1) + \Omega(\boldsymbol{w}^2) \big) + \lambda_2 \left( \Omega(\mathbf{F}^1) + \Omega(\mathbf{F}^2) \right)
\end{aligned} \tag{2.9}
$$

where $\mathcal{D}$ stands for our dataset, $N$ for the number of training samples, $\lambda_1$ is the regularization parameter corresponding to the linear feature effects and $\lambda_2$ is the regularization parameter targeting polyadic interactions.

### 2.5.3 Experiment Setup

The version of LINCS data we used comprise $22,412$ drugs applied to $56$ different tissues for $978$ landmark genes. We followed a standard protocol for data pre-processing, which is described in the Supplementary Material. We incorporate external features for the drug and gene data domains. For drugs, we use the substructure fingerprints denoting the chemical structure for each drug [78]. For genes, we use gene-gene similarity features computed using the GOSemSim [79] package in R. In all our experiments, we used the $10$ tissues containing the most expression profiles. We also discarded $128$ genes for which we did not have any similarity information.

We employ the holdout method to tune the hyper-parameters, so the available samples are split to train, validation and test sets with an approximate ratio of $0.6 : 0.2 : 0.2$. The validation set was purely used to tune the hyper-parameters, which are found using logarithmic grid search for each one of the methods. The hyper-parameters achieving the best performance on the validation set were selected and the corresponding performance on the test set is reported. We used $\ell_2$-norm regularization for all the methods.

We used the Spearman's $\rho$ (Spearman rank correlation coefficient ranging from $-1$ to $1$) between the vector of predicted and that of true measurements, as the measure of accuracy. This is a standard evaluation metric for gene expression analysis, where correlation metrics are usually preferred over error measures [68].

Polyadic Regression was set to compete with the following approaches:

**Ridge Regression.** A linear regression with L2 regularization without considering interaction terms. We used the *GLMNet* package [80] (Matlab version) implementing Ridge Regression.

**Factorization Machines (FMs)** [71]. FMs are efficiently exploring all the pairwise feature interactions. We used the *libFM* package [81] (C++) implementing FMs and the Monte Carlo Markov Chain (MCMC) fitting algorithm, which is recommended by the author as the least prone to hyper-parameter selection. Thus, other than the rank parameter governing the model size, we only had to tune the standard deviation of the initial distribution of parameters. Note that the regularization-related parameters are automatically determined in MCMC.

**Multi-view machines (MVMs)** [82]. This is another prior work which takes into account both linear and inter-domain interaction terms, and factorizes all of them jointly. We used the *zen* package implementing Multi-view Machines [83, 82] on top of Apache Spark (in local mode) using Scala. The fitting algorithm in this case is Gradient Descent with adaptive gradient (AdaGrad) [84], which is also recommended by the authors in [82], so that the algorithm is insensitive to the choice of the learning rate. We verified the performance

boost using AdaGrad and the insensitivity to the initial learning rate in our experiments when using it. Thus, apart from the rank parameter defining the model size, the only hyper-parameter we had to tune was the $\lambda$ value corresponding to $\ell_2$-norm regularization. Note that due to the joint factorization of all the parameters in MVMs, there is a single regularization value to cover the needs of both the linear and interaction terms.

To adapt to the supervised learning setting employed by Ridge Regression and FMs, for each sample $\{(i_1, i_2, i_3), y_{i_1,i_2,i_3}\}$, we create a "concatenated" feature vector $[\boldsymbol{x}_{i_1}^1; \boldsymbol{x}_{i_2}^2; \boldsymbol{e}_{i_3}^3] \in \mathbb{R}^{d_1+d_2+n_3}$.

Our method is implemented in Matlab R2015b and C++ with multi-threading capabilities (OpenMP). We used the Mex interface to bridge Matlab and C++ implementation. We also employed the Matlab package *apg* in [85], implementing an accelerated proximal gradient method.

We set all methods other than Ridge Regression to run for a maximum of $2,000$ iterations [3]. We do employ early-stopping through cross-validation to avoid overfitting for all methods.

In terms of hardware, we used a server running Ubuntu 14.04 with 251 GB of RAM and 16 physical Intel(R) Xeon(R) E5-2630 CPU's with a maximum clock frequency of 2.40GHz. Each one of the physical cores can exploit 2 threads with hyper-threading enabled.

### 2.5.4 Predicting missing polyadic data

In the following, we evaluate the accuracy of the methods under comparison in predicting missing measurements. Besides picking the top-10 tissues containing the most data, we sub-selected the drugs with measurements available in all of them. Thus, we have $81$ drugs for this setting. We can consider that the objects under consideration form a dense tensor containing $81 \times 850 \times 10$ elements denoting the combinations of drugs $\times$ genes $\times$ tissues.

[3] Ridge regression implementation does not require an iteration parameter.

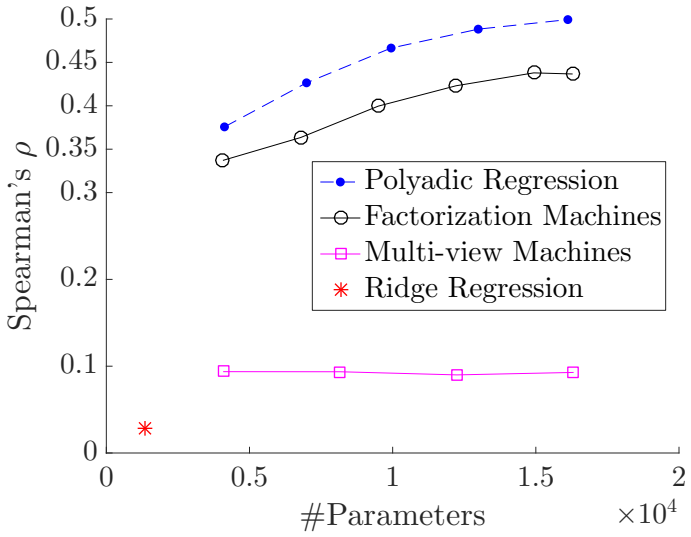



Figure 2.2: Predicting missing measurements: Spearman correlation between the predicted and the true vector of measurements, as we vary the model size.

Those $688,500$ expression values are split into train, validation and test sets, as explained in Section 2.5.3. Moreover, we cleaned the initial $881$ drug structure features to remove the ones without any variation among the drugs selected. Thus, for this setting, we have $497$ drug features kept. The number of gene features is the same as the number of genes since we construct a gene-gene similarity matrix.

We vary the number of parameters for each method (apart from Ridge Regression where the number of parameters is fixed to be the number of features), by tuning the corresponding rank-related parameters. Note that in Polyadic Regression, we have two parameters governing the number of parameters ($m_1$ and $m_2$, as shown in Equation 2.8). We consider that $m_1 = m_2$ and vary them in the range $\{2, 4, 6, 8, 10\}$. Accordingly, we vary the rank-related parameters in FMs and MVMs to reach comparable model sizes.

We present the results of those experiments in Figure 2.2. We notice that the output of Ridge Regression achieves almost zero correlation with the true measurements. This indicates that the linear terms do not have high predictive towards the predicted outcomes. This is perfectly reasonable considering for instance a specific drug feature (e.g., existence of a certain chemical bond). This feature may target a specific tissue on a certain group

of genes, but intuitively there are no drug features that have treating capabilities for every condition. Moreover, the MVMs do not achieve high accuracy either. This approach comes with an efficient model representation which has shown success in recommender systems applications [82]. However, it assumes that the linear and interaction terms are factorized jointly; thus, a single regularization parameter has to cover the needs of both the linear and interaction terms, even if the former ones are almost irrelevant towards the predicted measurements. We believe this is the reason behind the low accuracy of MVMs for this task. Finally, the FMs are promising for this task and the correlation achieved between the output and the measurements' vector reaches a maximum of $0.44$. However, they do not take into account $3$-order interactions, limiting the model expressiveness for Polyadic Prediction problems.

Polyadic Regression achieves the best performance for all model sizes, reaching a maximum of $0.5$ correlation between the output and the true measurements' vector. Note that maximum performance on the validation set was in most cases achieved by setting $\lambda_1$ (in equation 2.9) to be orders of magnitude larger than $\lambda_2$, where the former shrinks the linear and the latter regularizes the interaction effects. Thus, the flexibility of independent representation (and regularization) between linear and interaction terms is crucial towards the target task.

### 2.5.5 Predicting polyadic data for new drugs

We also tackle another important challenge in the field of chemogenomics: predicting expression values for new drugs, unseen during the training phase. First, we want to incorporate more drugs than the ones used in Section 2.5.4, so that the task becomes more challenging. Thus, instead of constraining the drugs to have relevant data in all the $10$ tissues, we require that they have data in at least $2$ out of them. This requirement is fulfilled by $2,169$ drugs, which we further randomly sub-sample to $500$. The number of relevant drug features (non-zero variation) is $612$ in this case. In sum, for the current task, we consider

Table 2.1: Predicting measurements for new drugs: for roughly the same model size, Polyadic Regression achieves $0.1$ increase in Spearman correlation between the predicted and the true vector of measurements, against the best-performing competing method. Results are averaged over $5$ runs and standard deviation is reported.

| Method | Spearman's $\rho$ | #Parameters |
|---|---|---|
| Polyadic Regression | $0.23025 \pm 0.0063886$ | 4471 |
| Factorization Machines | $0.1252 \pm 0.0083942$ | 4417 |
| Multi-view Machines | $0.0669 \pm 0.017242$ | 4425 |
| Ridge Regression | 0.0061 | 1473 |

$500$ drugs, $850$ genes and $10$ tissues. In contrast to Section 2.5.4, we only have $\approx 44\%$ of the measurements among the object combinations available, thus leading to $1,870,850$ data samples. Since we are only interested in predictions for new drugs (predicting for new genes is not a practical use case and we have no external knowledge for tissues), we follow the holdout method by constraining though that the the train, validation and test sets have no common drugs.

We evaluate the robustness of the competing approaches, thus we fixed Polyadic Regression, FMs and MVMs to roughly the same model size and run the experiments $5$ times, reporting average performance and standard deviation. We provide the results in Table 2.1. Polyadic Regression achieves $0.1$ increase in correlation between the predicted and the true vector of measurements, against the best-performing competing method. Moreover, we remark its robustness in terms of different initialization of parameters.

### 2.5.6 Scalability

We assess the scalability properties of the methods under comparison w.r.t. varying sizes of training data and input features. To do so, we used the data described in Section 2.5.5. Regarding the scalability for increasing number of training examples, we constructed smaller sets of data, by including random subsets of $1/8, 1/4, 1/2$ of the total samples. As concerns the scalability for increasing number of input features, we fixed the number of training examples and removed an equal number of features from the drug and gene domains, so as

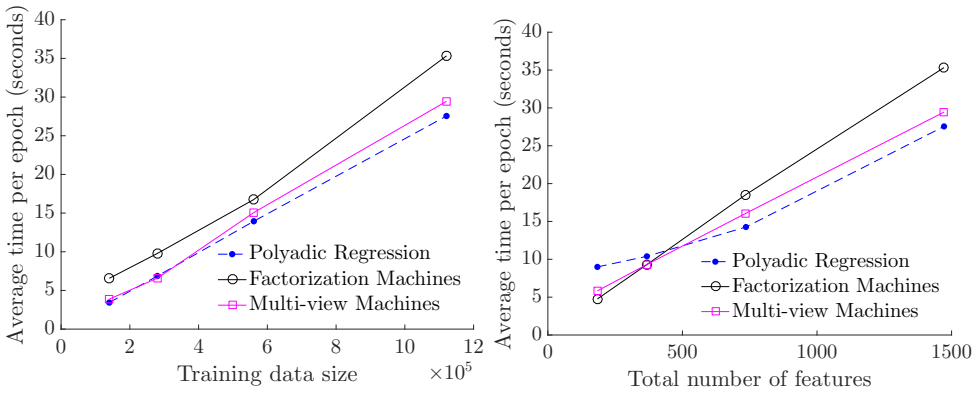



Figure 2.3: The scalability in terms of training time per epoch with respect to increasing training data size and input feature dimension.

to reach the desired total feature number. We measured the training time per epoch, as an average over 100 epochs.

Note that the *GLMNet* package implementing Ridge Regression uses a cyclical coordinate descent algorithm, updating a specific parameter in each iteration. This is in contrast to the rest of the methods which update the whole space of parameters; thus, a comparison in terms of time per epoch with Ridge Regression would not be meaningful and is not included. However, we empirically remark that it is the fastest method, which is reasonable considering it only takes into account linear feature effects.

We provide the results in Figure 2.3. We remark that all methods share similar (near-linear) scalability properties. We would like to emphasize that a direct time comparison between different methods would not be fair, since they are implemented using different setup (single-threaded, multi-threaded).

## 2.6 Related Work

Polyadic Regression essentially tackles the general case of the Dyadic Prediction problem which is studied in [65, 86, 66, 67]. The corresponding approaches though, only predict dyadic and cannot generalize to predicting polyadic data. Other works that also model only dyadic interactions are the Sparse Factorization Machine [87] and the Conditional

High-Order Boltzmann Machine [88].

Multi-view Machines [82] (MVMs) is a recently proposed model, exhibiting success in recommender systems' applications. MVMs capture both linear and higher-order interactions across features of different data domains, and *jointly* factorize all of them by a CP tensor decomposition [89]. On one hand, this decision limits the number of parameters to learn. However, it restricts to a target model where the linear and interaction terms are composed from the same low-rank factors, and have to share the same regularization. This may be a limiting factor when the interaction and linear terms have very different contributions towards the predicted measurements. Our model is more flexible, allowing for a different treatment among the linear and interaction terms.

As concerns other lines of work, the notion of multi-view learning, as it has been hitherto used in the literature [90], does not tackle the challenges we introduced in Section 2.1. It is limited to models either accepting only two inputs [91] or learning correlations at the view-level (as in multiple-kernel learning [92]) and not between features of different representations, thus limiting the model's expressiveness and interpretation potential. On a different note, while tensor regression methods (e.g., [93]) predict polyadic data by generally exploring only the highest order of possible interactions, they cannot provide predictions for new objects (e.g., new drugs in our target application), in the post-training phase.

## 2.7 Conclusions

We proposed Polyadic Regression, a general framework predicting measurements associated with multiple objects. Our framework enables predictions for new objects, unseen during training, thus tackling the so-called cold-start problem. Our model is expressive for addressing general Polyadic Prediction problems, by exploring all the high-order interactions across different data domains, in an efficient, factorized, way. We evaluate our approach with real chemogenomics data, demonstrating its superior accuracy over the prior art. As future work, we plan to apply Polyadic Regression to other fields and further im-

prove its scalability.

# CHAPTER 3

# SUSTAIN: SCALABLE UNSUPERVISED SCORING FOR TENSORS AND ITS APPLICATION TO PHENOTYPING


Summary

In this Chapter, we focus on extracting integer factors, out of co-occurrence input tensors derived from EHR data, which can be interpreted to identify patient groups sharing common traits (aka unsupervised phenotyping). Prior work applying and extending tensor methods for phenotyping focuses on real-valued models, which may need arbitrary hard thresholding to the ranked list of factor elements to achieve concise results. Also, assessing the relative frequency between different real-valued factor elements may be non-intuitive. To tackle these challenges, we introduce SUSTain, an integer-constrained tensor factorization framework, which shows either significantly better fit or orders of magnitude speedups for a comparable fit against several baselines and can be used to extract concise and clinically-meaningful phenotype definitions (87% of them were validated as meaningful by a cardiologist).


This chapter introduces a new method, which we call SUSTain, that extends real-valued matrix and tensor factorizations to data where values are integers. Such data are common when the values correspond to event counts or ordinal measures. The conventional approach is to treat integer data as real, and then apply real-valued factorizations. However, doing so fails to preserve important characteristics of the original data, thereby making it hard to interpret the results. Instead, our approach extracts factor values from integer datasets as *scores* that are constrained to take values from a small integer set. These scores are easy to interpret: a score of zero indicates no feature contribution and higher scores indicate *distinct levels* of feature importance.

At its core, SUSTain relies on: a) a problem partitioning into integer-constrained subproblems, so that they can be optimally solved in an efficient manner; and b) organizing the order of the subproblems' solution, to promote reuse of shared intermediate results. We propose two variants, $SUSTain_M$ and $SUSTain_T$, to handle both matrix and tensor inputs, respectively. We evaluate SUSTain against several state-of-the-art baselines on both synthetic and real Electronic Health Record (EHR) datasets. Comparing to those baselines, SUSTain shows either significantly better fit or orders of magnitude speedups that achieve a comparable fit (up to $425\times$ faster). We apply SUSTain to EHR datasets to extract patient phenotypes (i.e., clinically meaningful patient clusters). Furthermore, $87\%$ of them were validated as clinically meaningful phenotypes related to heart failure by a cardiologist.

## 3.1 Introduction

Matrix and tensor factorization are among the most promising approaches to extracting meaningful latent structure from multi-aspect data. They have been applied successfully in diverse applications, including social network analysis [15], text mining [16], image processing [17], recommendation systems [18], brain data analysis [19] and healthcare analytics [21], to name a few. Factorization models decompose input data into real-valued representatives revealing clusters with distinct interpretable feature profiles.

However, a significant problem arises when the input data are most naturally expressed as integer values. Examples include event counts and ordinal data [94]. In such cases, real-valued factors distort the original integer characteristics. For example, real values might no longer be interpretable as counts or frequencies. Also, the possible ranges and relative differences of elements in real-valued factors is arbitrary; this makes it hard to intuitively compare the importance of different elements. Furthermore, in many applications, practitioners are accustomed to interpreting integer-valued scores in standardized scales. Real-valued factors might require arbitrary thresholding or other unnatural transformations to convert into such scales, thereby inhibiting interpretation by domain experts.

A specific motivating application for our methods is clinical phenotyping from Electronic Health Records (EHR) data. Consider that a disease, such as heart failure, is often heterogeneous in that patients differ by underlying pathophysiology and needs. That is, a disease is often comprised of distinct disease subtypes, or phenotypes, which vary by the ensemble of causes, associations with other diseases, and treatment needs [38]. Phenotyping is intended to distinguish the latent structure among features that can, in turn, be used to prevent disease subtypes and improve treatment development and management [95]. EHR data offer a diverse and rich set of features (e.g., diagnostic, drug and procedure codes) that can serve to improve disease phenotyping. But, these data must often be represented in integer form (e.g., clinical event counts) to be utilized in unsupervised learning. For example, we can construct a patient-disease matrix where the $ij$-th element represents the number of times patient $i$ had disease $j$ documented in her records. Similarly, we can build higher-order tensors such as a patient-disease-medication one. The goal of unsupervised phenotyping is to identify patient clusters defined by unique feature sets, each one of which aligns with a distinct and intuitive clinical profile; in this work, we tackle this challenge via a scalable constrained integer tensor factorization.

Factorization methods have been successfully used for EHR-based unsupervised phenotyping [24, 21, 49, 50, 51]. In many of those settings, the problem can be formulated via Nonnegative Matrix Factorization (NMF) [17] e.g., minimizing the squared Frobenius norm of the error:

$$\min \left\{ ||\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V}^T||_F^2 \;\middle|\; \boldsymbol{U} \geq 0, \boldsymbol{V} \geq 0 \right\} \tag{3.1}$$

$\boldsymbol{X} \in \mathbb{Z}_+^{M \times N}$ is a non-negative integer input matrix whose $\boldsymbol{X}(i,j)$ cell reflects the event counts for the $i$-th (out of $M$) patient with respect to the $j$-th (out of $N$) features. Given an input number $R$ of desired phenotypes, the matrix $\boldsymbol{U} \in \mathbb{R}^{M \times R}$ corresponds to a membership matrix of the patients with respect to the $R$ phenotypes. And the matrix $\boldsymbol{V} \in \mathbb{R}^{N \times R}$ provides the phenotypes' definition: the non-zero elements of the $r$-th column $\boldsymbol{V}(:,r)$ reveal the potentially relevant features to the $r$-th phenotype.

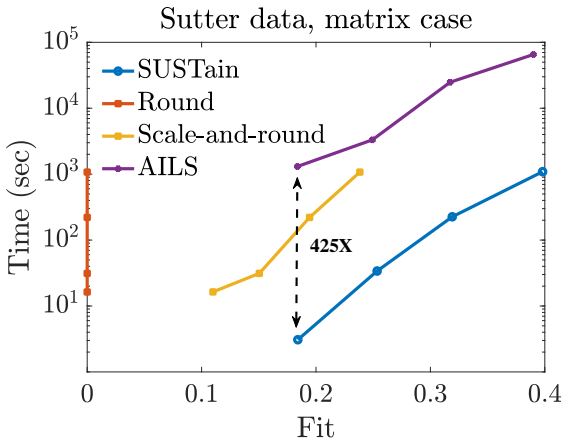



Figure 3.1: Fit (range $[0, 1]$) vs time trade-off for varying target number of phenotypes $R = \{5, 10, 20, 40\}$, on a patients-by-diagnoses matrix formed of $\approx$ 260K patients from Sutter Palo Alto Medical Foundation Clinics. $SUSTain_M$ is as accurate as the most accurate baseline (based on [96, 97, 94]), but up to $425\times$ faster ($R = 5$: $\approx 3$ seconds by $SUSTain_M$ vs. $\approx$ 22 minutes by AILS). Even for a larger target rank (e.g., $R = 20$), $SUSTain_M$ is $110\times$ faster ($\approx 4$ minutes by $SUSTain_M$ vs. $\approx 7$ hours by AILS). As compared to a carefully-designed heuristic that performs a scale-and-rounding of the real-valued solution, $SUSTain_M$ achieves up to $16\%$ higher fit. In summary, $SUSTain_M$ dominates all other baselines in both time and fit.

Interpreting those factors is crucial in order to determine whether and to what extent a patient exhibits a phenotype, as well as which set of candidate features should be considered to compose each $r$-th phenotype so that it is clinically meaningful. However, this can be challenging if the resulting factors contain arbitrary (nonnegative) real values. Real-valued factors distort the count nature of input data; thus, identifying cases and controls based on counts of relevant medical features [98, 57] is no longer possible. Also, the possible ranges and relative differences of elements in real-valued factors is arbitrary, thus impeding the practitioner's assessment of their relative importance. In practice, ad hoc heuristics have been introduced with limited success: a) hard thresholding to the ranked list of factor elements, which is usually arbitrary and leads to poor model fit; b) the factor values are hidden altogether and only the elements' ranking is preserved, which omits valuable information regarding the individual elements' actual importance.

**Contributions:** To tackle these challenges, we propose Scalable Unsupervised Scoring for Tensors (SUSTain), a framework extracting the factor values as *scores*, constrained to

Table 3.1: Most prevalent phenotype ($26\%$ of patients) extracted via $SUSTain_T$ for a heart failure cohort. The $r$-th phenotype prevalence is measured through the patient membership vectors containing non-zero element in the $r$-th coordinate. The score of each feature indicates its relative frequency. The prefix for each feature indicates whether it corresponds to a medication (Rx) or a diagnosis (Dx). The cardiologist labeled the result as "hyperlipidemia" and confirmed that the two features are clinically connected to heart failure.

| **Hyperlipidemia** | **Score** |
|---|---|
| Rx_HMG CoA Reductase Inhibitors | 3 |
| Dx_Disorders of lipid metabolism | 1 |

a small integer set. SUSTain offers a straightforward interpretation protocol: a score of zero indicates no feature contribution and higher scores indicate *distinct levels* of feature importance.

Our methodology relies on identifying a problem partitioning into integer-constrained sub-problems so that each one of them can be solved optimally in an efficient manner; at the same time, their solution order is organized so as to promote re-use of shared intermediate results. SUSTain can handle both matrix and tensor inputs, through $SUSTain_M$ and $SUSTain_T$ methods, which we formulate in Sections 3.3.1 and 3.3.2 respectively.

SUSTain yields faster and more scalable approaches than baselines achieving comparable fit, as evaluated on both synthetic (publicly-available) and real healthcare datasets. For example, as shown in Figure 3.1, $SUSTain_M$ achieves the same level of accuracy as the most accurate baseline **up to $\mathbf{425}\times$ faster**. $SUSTain_T$ can handle large-scale tensor inputs for which the most accurate baseline fails and scales linearly with the number of patients.

SUSTain's interpretation protocol is particularly meaningful for unsupervised phenotyping: it is easily understood by medical experts who are used to simple and concise, scoring-based descriptions of a patient's clinical status (e.g., risk scores[1]). While recent work derives risk scores for predictive modeling (supervised learning) [99], our application of SUSTain extracts scores for unsupervised phenotyping based on unlabeled EHR data. In

[1]In MDCalc, one can find a vast amount of such scores used in medicine.

Table 3.1, we provide a representative phenotype extracted through our method, as part of a case study we performed on phenotyping heart failure patients. The meaningfulness of the phenotype candidates extracted through this case study was confirmed by a cardiologist, who annotated $87\%$ of them as clinically meaningful phenotypes related to heart failure. We summarize our contributions as:

- **Scalable unsupervised scoring:** We propose SUSTain, a fast and scalable approach decomposing integer multi-aspect data into integer scores, preserving the original integer characteristics.
- **SUSTain can handle matrix and tensor input:** We present SUSTain for both matrix (Section 3.3.1) and tensor (Section 3.3.2) inputs, through $SUSTain_M$ and $SUSTain_T$ methods, respectively.
- **Evaluation on various datasets:** We evaluate both the matrix and tensor versions on both synthetic (publicly-available) and real healthcare datasets.
- **Phenotyping heart failure patients:** The interpretability of the extracted scoring-based phenotypes was confirmed by a cardiologist, who annotated $87\%$ of them as clinically meaningful.

To promote reproducibility, our code is open-sourced and publicly available at: `https://github.com/kperros/SUSTain`.

## 3.2 Background

In Table 3.2 we summarize the notations used throughout this work. Let $\boldsymbol{x_0} \in \mathbb{R}^n$. The *euclidean projection* of $\boldsymbol{x_0}$ to a set $C \subseteq \mathbb{R}^n$ is defined as $\Pi_C(\boldsymbol{x_0}) = \text{argmin}\,\{||\boldsymbol{x} - \boldsymbol{x_0}||_2^2 \mid \boldsymbol{x} \in C\}$; thus, it is the problem of determining the vector $\boldsymbol{x}^*$ among all $\boldsymbol{x} \in C$ which is the closest to $\boldsymbol{x_0}$ w.r.t. the Euclidean distance [100]. A matrix $\boldsymbol{X}$ is called *rank*-1 if it can be expressed as the outer product of 2 non-zero vectors: $\boldsymbol{X} = \boldsymbol{x} \circ \boldsymbol{y}$. The Khatri-Rao Product (KRP) [101] is

the "matching column-wise" Kronecker product: for two matrices $\boldsymbol{U} \in \mathbb{R}^{M\times R}$, $\boldsymbol{V} \in \mathbb{R}^{N\times R}$ their KRP is as follows: $\boldsymbol{U} \odot \boldsymbol{V} = [\boldsymbol{U}(:,1) \otimes \boldsymbol{V}(:,1) \;\; \boldsymbol{U}(:,2) \otimes \boldsymbol{V}(:,2) \;\; \ldots \;\; \boldsymbol{U}(:,R) \otimes \boldsymbol{V}(:,R)] \in \mathbb{R}^{MN\times R}$

A *tensor* is a multi-dimensional array. The tensor's *order* denotes the number of its dimensions, also known as ways or modes (e.g., matrices are 2-order tensors). A $d$-order tensor $\mathcal{X}$ is called *rank*-1 if it can be expressed as the outer product of $d$ non-zero vectors: $\boldsymbol{X} = \boldsymbol{a}_1 \circ \boldsymbol{a}_2 \circ \cdots \circ \boldsymbol{a}_d$. A *fiber* is a vector extracted from a tensor by fixing all modes but one. For example, a matrix column is a mode-1 fiber. A *slice* is a matrix extracted from a tensor by fixing all modes but two. *Matricization*, also called *reshaping* or *unfolding*, logically reorganizes tensors into other forms without changing the values themselves. The mode-$n$ matricization of a $d$-order tensor $\mathcal{X} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_d}$ is denoted by $\boldsymbol{X}_{(n)} \in \mathbb{R}^{I_n \times I_1 I_2 \ldots I_{n-1} I_{n+1} \ldots I_d}$ and arranges the mode-$n$ fibers of the tensor as columns of the resulting matrix. The Matricized-Tensor Times Khatri-Rao Product (MTTKRP) [102] w.r.t. mode-$n$ is the matrix multiplication $\boldsymbol{X}_{(n)}\ \boldsymbol{A}_{\odot}^{(-n)}$, where $\boldsymbol{A}_{\odot}^{(-n)}$ corresponds to the Khatri-Rao product of all the modes except the $n$-th. MTTKRP is the bottleneck operation in many sparse tensor algorithms.

Table 3.2: Notations used throughout this work.

| Symbol | Definition |
|---|---|
| $\mathcal{X}, \boldsymbol{X}, \boldsymbol{x}, x$ | Tensor, matrix, vector, scalar |
| $vec(\boldsymbol{X})$ | Vectorization operator for matrix $\boldsymbol{X}$ |
| $\Pi_C(\boldsymbol{x})$ | Euclidean projection of $\boldsymbol{x}$ to a set $C$ |
| $\boldsymbol{X}(:,i)$ | Spans the entire $i$-th column of $\boldsymbol{X}$ |
| $diag(\boldsymbol{x})$ | Diagonal matrix with vector $\boldsymbol{x}$ on the diagonal |
| $\boldsymbol{X}_{(n)}$ | mode-$n$ matricization of tensor $\mathcal{X}$ |
| $\boldsymbol{A}^{(n)}$ | factor matrix corresponding to mode $n$ |
| $\circ$ | Outer product |
| $\otimes$ | Kronecker product |
| $\odot$ | Khatri-Rao product |
| $\boldsymbol{A}_{\odot}^{(-n)}$ | Khatri-Rao product of all the factor matrices expect $\boldsymbol{A}^{(n)}$ |
| $\boldsymbol{M}^{(n)}$ | the MTTKRP corresponding to mode-$n$ |
| $*$ | Hadamard (element-wise) product |

## 3.3 The SUSTain framework

First we present SUSTain for matrix input. Then, we describe how SUSTain can be extended for general high-order tensor input. Finally, we provide our interpretation protocol of SUSTain for unsupervised phenotyping.

### 3.3.1 SUSTain for matrix input

**Model:** For an integer input matrix $\boldsymbol{X} \in \mathbb{Z}_+^{M \times N}$ and a certain target rank $R$, the problem can be defined as:

$$\min \left\{ ||\boldsymbol{X} - \boldsymbol{U}\ \boldsymbol{\Lambda}\ \boldsymbol{V}^T||_F^2 \;\middle|\; \boldsymbol{U} \in \mathbb{Z}_\tau^{M \times R}, \boldsymbol{V} \in \mathbb{Z}_\tau^{N \times R}, \boldsymbol{\Lambda} \in \mathbb{Z}_+^{R \times R} \right\} \tag{3.2}$$

where $\mathbb{Z}_\tau = \{0, 1, \ldots, \tau\}$ is the set of nonnegative integers up to $\tau$, $\mathbb{Z}_+ = \{1, 2, \ldots, \infty\}$ is the set of positive integers and $\boldsymbol{\Lambda}$ is a diagonal $R$-by-$R$ matrix. The above problem can be also formulated as $||\boldsymbol{X} - \sum_{r=1}^{R} \boldsymbol{\lambda}(r)\, \boldsymbol{U}(:, r)\, \boldsymbol{V}(:, r)^T||_F^2$ where $\boldsymbol{\lambda}(r) = \boldsymbol{\Lambda}(r, r)$. The reason for having $\boldsymbol{\lambda}(r)$ is to absorb any scaling of each $r$-th rank-1 component, since the entries of $\boldsymbol{U}$ and $\boldsymbol{V}$ factors are upper bounded by $\tau$. Note that the $\boldsymbol{\lambda}(r)$ values cannot be simply obtained through normalization as in the corresponding real-valued models (e.g., NMF [17, 103]), due to the integer constraints. Finally, note that the integer set $\mathbb{Z}_\tau$ can easily vary for different factor matrices and even allow negative integers; this can also happen for the input matrix $\boldsymbol{X}$. The formulation in Problem (3.2) favors simplicity of presentation and matches the need of phenotyping applications.

**Fitting Algorithm:** We employ an alternating updating scheme to tackle the non-convex optimization Problem (3.2). Our scheme leads to optimal solutions to each one of the subproblems in an efficient manner, while organizing the order of updates so as to promote re-use of already computed intermediate results.

We follow the intuition behind the Hierarchical Alternating Least Squares (HALS) framework, which enables isolating and solving for each $k$-th rank-1 component separately.

Thus, Problem (3.2) gives:

$$\min\{|| \boldsymbol{X} - \underbrace{\sum_{r=1, r\neq k}^{R} \boldsymbol{\lambda}(r)\, \boldsymbol{U}(:,r)\, \boldsymbol{V}(:,r)^T}_{\boldsymbol{R}_k} - \boldsymbol{\lambda}(k)\, \boldsymbol{U}(:,k)\, \boldsymbol{V}(:,k)^T ||_F^2 \tag{3.3}$$
$$| \boldsymbol{U} \in \mathbb{Z}_{\tau}^{M\times R}, \boldsymbol{V} \in \mathbb{Z}_{\tau}^{N\times R}, \boldsymbol{\Lambda} \in \mathbb{Z}_{+}^{R\times R}\}$$

where $\boldsymbol{R}_k$ corresponds to the "residual matrix" and is considered fixed when solving for the $k$-th rank-1 component. The objective can be written as [104]:

$$J = ||\boldsymbol{R}_k||_F^2 + \boldsymbol{\lambda}^2(k)\ ||\boldsymbol{U}(:,k)||_2^2\, ||\boldsymbol{V}(:,k)||_2^2 - 2\, \boldsymbol{\lambda}(k)\, \boldsymbol{U}(:,k)^T\, \boldsymbol{R}_k\, \boldsymbol{V}(:,k)$$

We set:

$$\partial J/\partial \boldsymbol{\lambda}(k) = 2\, \boldsymbol{\lambda}(k)\ ||\boldsymbol{U}(:,k)||_2^2\, ||\boldsymbol{V}(:,k)||_2^2 - 2\, \boldsymbol{U}(:,k)^T\, \boldsymbol{R}_k\, \boldsymbol{V}(:,k) = 0$$

and obtain:

$$\lambda_k^* := \frac{\boldsymbol{U}(:,k)^T\, \boldsymbol{R}_k\, \boldsymbol{V}(:,k)}{||\boldsymbol{U}(:,k)||_2^2\, ||\boldsymbol{V}(:,k)||_2^2}$$

If $\boldsymbol{U}(:,k)^T\, \boldsymbol{R}_k\, \boldsymbol{V}(:,k) > 0$ then the minimum value of $J$ for $\boldsymbol{\lambda}(k) \in \mathbb{Z}_+$ is obtained at $max(1, round(\lambda_k^*))$ where $round()$ rounds to the nearest integer. If $\boldsymbol{U}(:,k)^T\, \boldsymbol{R}_k\, \boldsymbol{V}(:,k) \leq 0$, then the minimum objective value for $\boldsymbol{\lambda}(k) \in \mathbb{Z}_+$ is attained at $\boldsymbol{\lambda}(k) = 1$. Combining these two cases, the optimal $\boldsymbol{\lambda}(k) \in \mathbb{Z}_+$ is given by:

$$\boldsymbol{\lambda}(k) \leftarrow max\left(1, round\left(\frac{\boldsymbol{U}(:,k)^T\, \boldsymbol{R}_k\, \boldsymbol{V}(:,k)}{||\boldsymbol{U}(:,k)||_2^2\, ||\boldsymbol{V}(:,k)||_2^2}\right)\right) \tag{3.4}$$

In practice, $\boldsymbol{R}_k$ may be large ($M \times N$) and dense, even if the input is sparse (as happens in our main motivating application); thus its explicit materialization should be avoided [105, 12]. Expanding the above expression gives:

$$\boldsymbol{\lambda}(k) \leftarrow max\left(1, round\left(\boldsymbol{\lambda}(k) + \frac{\boldsymbol{V}(:,k)^T\ \left([\boldsymbol{X}^T\, \boldsymbol{U}]_{:,k} - \boldsymbol{V}\, \boldsymbol{\Lambda}\, [\boldsymbol{U}^T\, \boldsymbol{U}]_{:,k}\right)}{[\boldsymbol{U}^T\, \boldsymbol{U}]_{k,k}\, [\boldsymbol{V}^T\, \boldsymbol{V}]_{k,k}}\right)\right) \tag{3.5}$$

Next, solving Problem (3.3) for $\boldsymbol{V}(:,k)$ gives:

$$\min \left\{ ||\boldsymbol{R}_k - \boldsymbol{\lambda}(k)\, \boldsymbol{U}(:,k)\, \boldsymbol{V}(:,k)^T||_2^2 \;\middle|\; \boldsymbol{V}(:,k) \in \mathbb{Z}_\tau^N \right\} \tag{3.6}$$

To solve the above, we apply the Optimal Scaling Lemma [106] for the integer constraint. This Lemma states that for any set $C$ of constraints imposed on $\boldsymbol{b}$, it holds that:

$$\min \left\{ ||\boldsymbol{Y} - \boldsymbol{x}\, \boldsymbol{b}^T||_2^2 \;\middle|\; \boldsymbol{b} \in C \right\} = \Pi_C(\boldsymbol{\beta})$$

where $\boldsymbol{\beta} = \frac{\boldsymbol{x}^T \boldsymbol{Y}}{\boldsymbol{x}^T \boldsymbol{x}}$ is the unconstrained solution to the above problem. This means that the optimal solution of the constrained problem is simply the projection of the unconstrained solution onto the constraint set $C$. Thus, the optimal solution of Problem (3.6) is:

$$\boldsymbol{V}(:,k) \leftarrow \boldsymbol{\Pi}_{\mathbb{Z}_\tau^N} \left( \frac{\boldsymbol{R}_k^T\, \boldsymbol{U}(:,k)}{[\boldsymbol{U}^T\, \boldsymbol{U}]_{k,k}\, \lambda(k)} \right) \tag{3.7}$$

Since $\mathbb{Z}_\tau^N$ is the Cartesian product of subsets of the real line, i.e., $\mathbb{Z}_\tau^N = \underbrace{\mathbb{Z}_\tau \times \mathbb{Z}_\tau \times \cdots \times \mathbb{Z}_\tau}_{N \text{ times}}$ we can take

$$\Pi_{\mathbb{Z}_\tau^N}(\boldsymbol{V}(:,k)) = [\Pi_{\mathbb{Z}_\tau}(\boldsymbol{V}(1,k)), \ldots, \Pi_{\mathbb{Z}_\tau}(\boldsymbol{V}(N,k))] \tag{3.8}$$

thus project each scalar coordinate individually. For a real-valued scalar $\alpha$, projecting onto $\mathbb{Z}_\tau$ gives [107]:

$$\Pi_{\mathbb{Z}_\tau}(\alpha) = min\left(max\left(round\left(\alpha\right), 0\right), \tau\right) \tag{3.9}$$

Finally, expanding $\boldsymbol{R}_k$ in Expression (3.7), combining with (3.8), (3.9) and setting:

$$\boldsymbol{b} \leftarrow \boldsymbol{V}(:,k) + \frac{[\boldsymbol{X}^T\, \boldsymbol{U}]_{:,k} - \boldsymbol{V}\, \boldsymbol{\Lambda}\, [\boldsymbol{U}^T\, \boldsymbol{U}]_{:,k}}{[\boldsymbol{U}^T\, \boldsymbol{U}]_{k,k}\, \boldsymbol{\lambda}(k)} \tag{3.10}$$

gives the optimal solution for Problem (3.6):

$$\boldsymbol{V}(:,k) \leftarrow min\left(max\left(round\left(\boldsymbol{b}\right), 0\right), \tau\right) \tag{3.11}$$

where $min(), max(), round()$ are taken element-wise.

Having derived the updates for $\boldsymbol{\lambda}(k)$, $\boldsymbol{V}(:,k)$, in Relations (3.5) and (3.11) respectively, we remark that the computationally expensive intermediate results $[\boldsymbol{X}^T\ \boldsymbol{U}]$ and $[\boldsymbol{U}^T\ \boldsymbol{U}]$ are shared between them. To exploit that, we choose to successively update $\boldsymbol{\lambda}(k)$ and $\boldsymbol{V}(:,k)$ during the same iteration and iterate $\forall k \in \{1,\ldots,R\}$. As a result of the proposed update order, the only non-negligible additional operation in order to compute both $\boldsymbol{\lambda}(k)$ and $\boldsymbol{V}(:,k)$ is to re-compute $\boldsymbol{t} := \boldsymbol{V}\ \boldsymbol{\Lambda}\ [\boldsymbol{U}^T\ \boldsymbol{U}]_{:,k}$ after having updated $\boldsymbol{\lambda}(k)$.

Re-computing $\boldsymbol{t}$ can be further optimized by observing that only the contribution of the $k$-th component $\boldsymbol{t_k} := \boldsymbol{V}(:,k)\ *\ \boldsymbol{\lambda}(k)\ [\boldsymbol{U}^T\ \boldsymbol{U}]_{k,k}$ has to be adjusted. Thus, we can store $\boldsymbol{t}$ and $\boldsymbol{t_k}$, compute $\boldsymbol{\lambda}'(k)$, and then adjust $\boldsymbol{t}$ as: $\boldsymbol{t} \leftarrow \boldsymbol{t} - \boldsymbol{t_k} + \big(\boldsymbol{V}(:,k)\ *\ \boldsymbol{\lambda}'(k)\ [\boldsymbol{U}^T\ \boldsymbol{U}]_{k,k}\big)$.

Updating $\boldsymbol{U}(:,k)$ can be executed in symmetric fashion to $\boldsymbol{V}(:,k)$. In Algorithm 1, we present our main procedure to update both the factor matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ and $\boldsymbol{\lambda}$ values in an alternating fashion. In Algorithm 2, we provide the definition of $SUSTain$_Update_Factor which updates a single factor (denoted as $\boldsymbol{F}$) and the vector $\boldsymbol{\lambda}$.

**Algorithm 1** $SUSTain_M$

**Input:** $\boldsymbol{X} \in \mathbb{R}^{M \times N}$, target rank $R$ and upper bound $\tau$
**Output:** $\boldsymbol{U} \in \mathbb{Z}_{\tau}^{M \times R}, \boldsymbol{V} \in \mathbb{Z}_{\tau}^{N \times R}, \boldsymbol{\lambda} \in \mathbb{Z}_{+}^{R}$
1: Initialize $\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{\Lambda}$
2: **while** convergence criterion is not met **do**
3: $\quad \boldsymbol{F} \leftarrow \boldsymbol{U}, \boldsymbol{M} \leftarrow \boldsymbol{X}\ \boldsymbol{V}, \boldsymbol{C} \leftarrow \boldsymbol{V}^T\ \boldsymbol{V}$
4: $\quad [\boldsymbol{U}, \boldsymbol{\lambda}] = SUSTain$_Update_Factor$(\boldsymbol{F}, \boldsymbol{M}, \boldsymbol{C}, \boldsymbol{\lambda}, R, \tau)$
5: $\quad \boldsymbol{F} \leftarrow \boldsymbol{V}, \boldsymbol{M} \leftarrow \boldsymbol{X}^T\ \boldsymbol{U}, \boldsymbol{C} \leftarrow \boldsymbol{U}^T\ \boldsymbol{U}$
6: $\quad [\boldsymbol{V}, \boldsymbol{\lambda}] = SUSTain$_Update_Factor$(\boldsymbol{F}, \boldsymbol{M}, \boldsymbol{C}, \boldsymbol{\lambda}, R, \tau)$
7: **end while**

**Computational Complexity:** The asymptotic cost of executing Algorithm 2 is $2R^2I$ flops (i.e., floating-point operations), $\forall R > 5$. This step costs $2R^2N$ when updating $\boldsymbol{V}$ and $2R^2M$ when updating $\boldsymbol{U}$. In Algorithm 1, assuming the input $\boldsymbol{X}$ is sparse, the cost of each one of $\boldsymbol{X}\ \boldsymbol{V}$ and $\boldsymbol{X}^T\ \boldsymbol{U}$ is $2\ nnz(\boldsymbol{X})\ R$ flops. Also, computing $\boldsymbol{V}^T\ \boldsymbol{V}$ and $\boldsymbol{U}^T\ \boldsymbol{U}$ cost $2R^2N$ and $2R^2M$ flops respectively. Thus, the total cost is: $4R\,(nnz(\boldsymbol{X}) + (M+N)R)$ flops.

**Algorithm 2** $SUSTain$_Update_Factor($\boldsymbol{F}, \boldsymbol{M}, \boldsymbol{C}, \boldsymbol{\lambda}, R, \tau$)

**Input:** $\boldsymbol{F} \in \mathbb{Z}_\tau{}^{I \times R}, \boldsymbol{M} \in \mathbb{R}^{I \times R}, \boldsymbol{C} \in \mathbb{R}^{R \times R}, \boldsymbol{\lambda} \in \mathbb{Z}_+^R$, target rank $R$ and upper bound $\tau$
**Output:** $\boldsymbol{F} \in \mathbb{Z}_\tau{}^{I \times R}, \boldsymbol{\lambda} \in \mathbb{Z}_+^R$

**for** $k = 1, \ldots, R$ **do**
  $\boldsymbol{t} \leftarrow \boldsymbol{F}\ (\boldsymbol{\lambda} * \boldsymbol{C}(:,k))$
  $\boldsymbol{t_k} \leftarrow \boldsymbol{F}(:,k)\ *\ \boldsymbol{\lambda}(k)\ \boldsymbol{C}(k,k)$
  $\alpha \leftarrow \boldsymbol{\lambda}(k) + \frac{\boldsymbol{F}(:,k)^T(\boldsymbol{M}(:,k) - \boldsymbol{t})}{\boldsymbol{C}(k,k)\ [\boldsymbol{F}^T\ \boldsymbol{F}]_{k,k}}$
  $\lambda(k) \leftarrow max\,(1, round\,(\alpha))$
  $\boldsymbol{t} \leftarrow \boldsymbol{t} - \boldsymbol{t_k} + (\boldsymbol{F}(:,k)\ *\ \boldsymbol{\lambda}(k)\ \boldsymbol{C}(k,k))$
  $\boldsymbol{b} \leftarrow \boldsymbol{F}(:,k) + \frac{\boldsymbol{M}(:,k) - \boldsymbol{t}}{C(k,k)\ \boldsymbol{\lambda}(k)}$
  $\boldsymbol{F}(:,k) \leftarrow min\,(max\,(round\,(\boldsymbol{b}), 0), \tau)$
**end for**

### 3.3.2 SUSTain for tensor input

**Model:** For a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots I_d}$ of order $d$ and a certain target rank $R$, the problem can be defined as

$$\min\{||\mathcal{X} - \sum_{r=1}^{R} \boldsymbol{\lambda}(r)\ \boldsymbol{A}^{(1)}(:,r)\ \circ \ldots\ \circ\ \boldsymbol{A}^{(d)}(:,r)||_F^2 \mid \boldsymbol{A}^{(n)} \in \mathbb{Z}_\tau^{I_n \times R}, \boldsymbol{\lambda}(r) \in \mathbb{Z}_+\} \quad (3.12)$$

where $n = \{1, \ldots, d\}, \mathbb{Z}_\tau = \{0, 1, \ldots, \tau\}$ is the set of nonnegative integers up to $\tau$ and $\mathbb{Z}_+ = \{1, 2, \ldots, \infty\}$ is the set of positive integers. Our model is an extension of $SUSTain_M$ presented in Section 3.3.1 for high-order tensors. It can be viewed as a constrained version of the CP tensor model [23, 89, 108].

**Fitting Algorithm:** Similarly to the matrix case, we set:

$$\mathcal{R}_k := \mathcal{X} - \sum_{r=1, r \neq k}^{R} \boldsymbol{\lambda}(r)\ \boldsymbol{A}^{(1)}(:,r)\ \circ \ldots\ \circ\ \boldsymbol{A}^{(d)}(:,r)$$

Thus, Problem (3.12) becomes:

$$\min\{||\mathcal{R}_k - \boldsymbol{\lambda}(k)\ \boldsymbol{A}^{(1)}(:,k)\ \circ \ldots\ \circ\ \boldsymbol{A}^{(d)}(:,k)||_F^2 \mid \boldsymbol{A}^{(n)} \in \mathbb{Z}_\tau^{I_n \times R}, \boldsymbol{\lambda}(k) \in \mathbb{Z}_+\} \quad (3.13)$$

We matricize the above expression w.r.t. mode-$n$ and utilize the fact that the mode-$n$ matricization of a rank-1 tensor $\boldsymbol{b}_1 \circ \cdots \circ \boldsymbol{b}_d$ can be expressed as $\boldsymbol{b}_n \left(\boldsymbol{b}_d \otimes \cdots \otimes \boldsymbol{b}_{n+1} \otimes \boldsymbol{b}_{n-1} \otimes \cdots \otimes \boldsymbol{b}_1\right)^T$ [109]

$$\begin{aligned}\min\{||\mathcal{R}_{k(n)} - \boldsymbol{\lambda}(k)\, \boldsymbol{A}^{(n)}(:,k) \\ (\boldsymbol{A}^{(d)}(:,k) \otimes \cdots \otimes \boldsymbol{A}^{(n+1)}(:,k) \otimes \boldsymbol{A}^{(n-1)}(:,k) \otimes \cdots \otimes \boldsymbol{A}^{(1)}(:,k))^T||_F^2 \\ | \ \boldsymbol{A}^{(n)} \in \mathbb{Z}_\tau^{I_n \times R}, \boldsymbol{\lambda}(k) \in \mathbb{Z}_+\}\end{aligned} \quad (3.14)$$

We set $\boldsymbol{A}_{\odot}^{(-n)} := \boldsymbol{A}^{(d)} \odot \cdots \odot \boldsymbol{A}^{(n+1)} \odot \boldsymbol{A}^{(n-1)} \odot \cdots \odot \boldsymbol{A}^{(1)}$ as the Khatri-Rao Product of all the factor matrices except the $n$-th and

$$\boldsymbol{C}^{(-n)} := \boldsymbol{A}^{(d)^T}\, \boldsymbol{A}^{(d)} * \cdots * \boldsymbol{A}^{(n+1)^T}\, \boldsymbol{A}^{(n+1)} * \boldsymbol{A}^{(n-1)^T}\, \boldsymbol{A}^{(n-1)} * \cdots * \boldsymbol{A}^{(1)^T}\, \boldsymbol{A}^{(1)} \quad (3.15)$$

as the Hadamard product of the Gram matrices of all the factor matrices except the $n$-th. Then, Objective (3.14) becomes

$$\min\{||\mathcal{R}_{k(n)} - \boldsymbol{\lambda}(k)\, \boldsymbol{A}^{(n)}(:,k)\, \boldsymbol{A}_{\odot}^{(-n)}(:,k)^T||_F^2 |\ \boldsymbol{A}^{(n)} \in \mathbb{Z}_\tau^{I_n \times R}, \boldsymbol{\lambda}(k) \in \mathbb{Z}_+\} \quad (3.16)$$

Solving the above for $\boldsymbol{\lambda}(k)$ can be handled equivalently to the corresponding matrix case (Relation (3.4)). Thus, the optimal solution for $\boldsymbol{\lambda}(k) \in \mathbb{Z}_+$ is:

$$\boldsymbol{\lambda}(k) \leftarrow max\left(1, round\left(\frac{\boldsymbol{A}^{(n)}(:,k)^T\, \mathcal{R}_{k(n)}\, \boldsymbol{A}_{\odot}^{(-n)}(:,k)}{||\boldsymbol{A}^{(n)}(:,k)||_2^2\, ||\boldsymbol{A}_{\odot}^{(-n)}(:,k)||_2^2}\right)\right) \quad (3.17)$$

By exploiting that [11]:

$$||\boldsymbol{A}_{\odot}^{(-n)}(:,k)||_2^2 = \boldsymbol{A}_{\odot}^{(-n)}(:,k)^T \boldsymbol{A}_{\odot}^{(-n)}(:,k) = [\boldsymbol{A}_{\odot}^{(-n)^T}\, \boldsymbol{A}_{\odot}^{(-n)}]_{k,k} = \boldsymbol{C}^{(-n)}(k,k)$$

and expanding:

$$\mathcal{R}_{k(n)}\, \boldsymbol{A}_{\odot}^{(-n)}(:,k) = \boldsymbol{M}^{(n)}(:,k) - \boldsymbol{A}^{(n)}\, \boldsymbol{\Lambda}\, \boldsymbol{C}^{(-n)}(:,k) + \boldsymbol{\lambda}(k)\, \boldsymbol{C}^{(-n)}(k,k)\, \boldsymbol{A}^{(n)}(:,k) \quad (3.18)$$

where $\boldsymbol{M}^{(n)}(:,k)$ is the Matricized-Tensor Times Khatri-Rao Product (MTTKRP) [102]

operation w.r.t. mode $n$, we get the optimal solution for $\boldsymbol{\lambda}(k) \in \mathbb{Z}_+$ as:

$$\boldsymbol{\lambda}(k) \leftarrow max\left(1, round\left(\boldsymbol{\lambda}(k) + \frac{\boldsymbol{A}^{(n)}(:,k)^T\left(\boldsymbol{M}^{(n)}(:,k) \; - \; \boldsymbol{A}^{(n)}\,\boldsymbol{\Lambda}\,\boldsymbol{C}^{(-n)}(:,k)\right)}{\boldsymbol{C}^{(-n)}(k,k)\,[\boldsymbol{A}^{(n)^T}\boldsymbol{A}^{(n)}]_{k,k}}\right)\right) \tag{3.19}$$

Next, we transpose the Objective (3.16) and solving for $\boldsymbol{A}^{(n)}(:,k)$ can be handled as in the matrix case (Relation (3.7)) through the Optimal Scaling Lemma [106]. Thus, the optimal $\boldsymbol{A}^{(n)}(:,k) \in \mathbb{Z}_\tau^{I_n}$ is given by:

$$\boldsymbol{A}^{(n)}(:,k) \leftarrow \boldsymbol{\Pi}_{\mathbb{Z}_\tau^{I_n}}\left(\frac{\mathcal{R}_{k(n)}\,\boldsymbol{A}_{\odot}^{(-n)}(:,k)}{\boldsymbol{C}^{(-n)}(k,k)\,\lambda(k)}\right) \tag{3.20}$$

Finally, combining Equation (3.18) into the above gives:

$$\boldsymbol{A}^{(n)}(:,k) \leftarrow \boldsymbol{\Pi}_{\mathbb{Z}_\tau^{I_n}}\left(\boldsymbol{A}^{(n)}(:,k) + \frac{\boldsymbol{M}^{(n)}(:,k) - \boldsymbol{A}^{(n)}\,\boldsymbol{\Lambda}\,\boldsymbol{C}^{(-n)}(:,k)}{\boldsymbol{C}^{(-n)}(k,k)\,\boldsymbol{\lambda}(k)}\right) \tag{3.21}$$

Note the direct correspondence of the above formulations for $\boldsymbol{\lambda}(k)$, $\boldsymbol{A}^{(n)}(:,k)$ with the core update Algorithm 2 we used for the matrix case. If we set $\boldsymbol{F} \leftarrow \boldsymbol{A}^{(n)}, \boldsymbol{M} \leftarrow \boldsymbol{M}^{(n)}, \boldsymbol{C} \leftarrow \boldsymbol{C}^{(-n)}$ then we can simply use Algorithm 2 to update a single factor $\boldsymbol{A}^{(n)}$ and the $\boldsymbol{\lambda}$ values. Also, we can exploit the development of existing scalable software libraries computing the bottleneck MTTKRP kernel for sparse data efficiently [102]. In Algorithm 3, we summarize the operations of our methodology for tensor input.

**Algorithm 3** $SUSTain_T$

**Input:** $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots I_d}$, target rank $R$ and upper bound $\tau$
**Output:** $\boldsymbol{A}^{(n)} \in \mathbb{Z}_\tau^{I_n \times R}$, with $n \in \{1, \ldots, d\}, \boldsymbol{\lambda} \in \mathbb{Z}_+^R$

```
Initialize A^(n), λ
while convergence criterion is not met do
    for n = 1, ..., d do
        M^(n) ← X_(n) A_⊙^(-n)                    // MTTKRP
        Compute C^(-n) as in Relation (3.15)
        [A^(n), λ] = SUSTain_Update_Factor(A^(n), M^(n), C^(-n), λ, R, τ)
    end for
end while
```

**Computational Complexity:** Updating the $n$-th mode in Algorithm 3 requires: $3\,R\,nnz(\mathcal{X})$ flops to compute the MTTKRP using state-of-the-art libraries for sparse tensors [102],

$2\ R^2\ I_n$ flops to compute $\boldsymbol{A}^{(n)^T}\ \boldsymbol{A}^n$ and $(d-1)\ R^2$ flops to update $\boldsymbol{C}^{(-n)}$ as in Equation (3.15). As discussed in the matrix case, the dominant cost of Algorithm 2 is $2\ R^2\ I_n$ flops. Overall, Algorithm 3 requires: $3\ d\ R\ nnz(\mathcal{X}) + 4\ R^2\ \sum_{n=1}^{d} I_n + d\ (d-1)\ R^2$ flops. In our experiments, the first term, thus the computation of MTTKRP, dominates the total cost.

### 3.3.3 Interpretation for phenotyping

Given the EHRs of a certain cohort, we form a patient-by-diagnoses matrix $\boldsymbol{X}$, whose $\boldsymbol{X}(i,j)$ cell is the number of encounters of patient $i$ where encounter diagnosis $j$ was recorded. In that case, the patient membership vector $\boldsymbol{U}(i,:)$ of $SUSTain_M$ provides the distinct levels of frequency of each one of the $R$ phenotypes throughout the medical history of the $i$-th patient. Likewise, each column $\boldsymbol{V}(:,r)$ indicates the frequency levels of each medical feature w.r.t. the $r$-th phenotype. Table 3.1 summarizes a phenotype example that accounts for the largest share of heart failure patients. Finally, due to the integer box (i.e., $\{0,\ldots,\tau\}$) constraints employed on the factor matrices, we can interpret the integer $\boldsymbol{\lambda}(r)$ values as scaling up the input encounter counts for the $r$-th phenotype. Thus, phenotypes with higher $\boldsymbol{\lambda}(r)$ values are expected to describe more persistent medical conditions, with higher number of associated encounters.

The above interpretation can be extended to the tensor case. Consider a tensor $\mathcal{X}$ whose $\mathcal{X}(i,j,k)$ cell defines the count of encounters of patient $i$ where medication $k$ was ordered for the patient with diagnosis $j$ as the order indication. Factorizing this tensor using $SUSTain_T$ yields a patient factor $\boldsymbol{A}^{(1)}$ which can be interpreted similarly to the $\boldsymbol{U}$ factor in the matrix case. Also, the factor matrices $\boldsymbol{A}^{(2)}, \boldsymbol{A}^{(3)}$ corresponding to diagnosis and medication or procedure phenotypes can be interpreted similarly to the $\boldsymbol{V}$ factor in the matrix case. The same applies to the $\boldsymbol{\lambda}(r)$ model values.

## 3.4 Experiments

### 3.4.1 Setup

*Description of datasets*

Table 3.3 summarizes statistics for the datasets used.

**Sutter:** This dataset corresponds to EHRs from Sutter Palo Alto Medical Foundation (PAMF) Clinics. The patients are 50 to 80 years old adults chosen for a heart failure study [110]. To form a patient-by-diagnosis matrix input, we extracted the number of encounter records with a specific diagnosis for each patient. To form a patient-diagnosis-medication tensor input, we used the medication orders, reflecting the ordered medications and the indicated diagnosis. We adopt standard medical concept groupers to group the available ICD-9 diagnosis codes [43] into Clinical Classification Software (CCS) [111] diagnostic categories (level 4). We also group the normalized drug names (i.e., combining all branded names and the generic name for a medication) based on unique therapeutic subclasses using the Anatomical Therapeutic Chemical Classification System [112].

**CMS:** We used a *publicly-available* CMS Linkable 2008-2010 Medicare Data Entrepreneurs' Synthetic Public Use File (DE-SynPUF) [2] that contains three years of claim records synthesized (i.e., to protect privacy) from $5\%$ of the 2008 Medicare population. CMS creates twenty $5\%$ subsamples of the claims data. We used the carrier claims data available from DE-SynPUF for the patients belonging to Samples 1 & 2. We increase the number of samples (i.e., number of patients) considered for the experiments related to assessing scalability. We used the diagnostic code information to build the input matrix and the diagnoses and procedures recorded to build the input tensor. In particular, we group the available ICD-9 diagnosis codes [43] into CCS [111] diagnostic categories (level 4) and use the CCS flat code grouper [111] to transform the CPT procedure codes available into procedure

[2] These data can be downloaded from `https://www.cms.gov/Research-Statistics-Data-and-Systems/Downloadable-Public-Use-Files/SynPUFs/DE_Syn_PUF.html`

Table 3.3: For each dataset used, we list its name, nature of input modes, their sizes and the approximate number of non-zeros. Pat refers to patients, Dx to diagnoses, Rx to medications and Proc to procedures.

| dataset | modes | size of modes | #nnz (≈Millions) |
|---|---|---|---|
| Sutter-matrix | Pat-Dx | $259{,}999 \times 576$ | 5.7 |
| Sutter-tensor | Pat-Dx-Rx | $248{,}347 \times 552 \times 555$ | 5.4 |
| CMS-matrix | Pat-Dx | $197{,}212 \times 583$ | 10.9 |
| CMS-tensor | Pat-Dx-Proc | $197{,}143 \times 583 \times 239$ | 23.4 |

categories.

*Baselines*

Below, we describe our efforts to design competitive baseline methods producing the target models in Problems 3.2 and 3.12, for the matrix and the tensor cases respectively.

**Round:** This baseline rounds the factor matrices from nonnegative matrix/tensor factorization. In the matrix case, we used the implementation of Nonnegative Matrix Factorization (NMF) [12, 113] and projected all the entries of the resulting factor matrices to $\mathbb{Z}_\tau$. We also set $\boldsymbol{\lambda}$ to an all-ones vector as NMF typically does not have the diagonal matrix $\boldsymbol{\Lambda}$. A typical issue of naively rounding NMF solutions is that values that are lower than $0.5$ are rounded to $0$, so a potentially large part of model information can be lost.

In the tensor case, we used the CP-ALS algorithm as in the Tensor Toolbox [114], adjusted to impose non-negativity constraints [113] on the factor matrices. Also, in contrast to the NMF case, CP-ALS produces a $\boldsymbol{\lambda}$ vector of nonnegative real values. In order to alleviate the effect of zeroing out values less than $0.5$ we compute the cube root of the $\boldsymbol{\lambda}$ vector element-wise and form a vector $\hat{\boldsymbol{\lambda}}$. Then, we absorb this scaling in the factor matrices by multiplying $\boldsymbol{A}^{(n)}\, diag(\hat{\boldsymbol{\lambda}}), \forall n = \{1, \ldots, d\}$ where $d$ is the input tensor's order. Finally, we set $\boldsymbol{\lambda}$ to an all-ones vector and project all the entries of the resulting factor matrices to $\mathbb{Z}_\tau$.

**Scale-and-round:** We design a more sophisticated scale-and-rounding heuristic which scales the factor matrices of the real-valued solutions before performing the rounding. This

step further alleviates the problem of zeroing out values less than $0.5$.

In the matrix case, we define the scaling factor $\gamma_2(j) = \tau/max(\boldsymbol{V}(:,j))$. Then, $\tilde{\boldsymbol{V}}(:,j)$ $= round\,(\gamma_2(j)\boldsymbol{V}(:,j))$. Similarly, we define $\gamma_1(j) = \tau\ /max(\boldsymbol{U}(:,j))$. Then, $\tilde{\boldsymbol{U}}(:,j)$ $= round\,(\gamma_1(j)\boldsymbol{U}(:,j))$. Those steps scale-up the maximum value of each factor matrix column to reach the upper bound $\tau$. Then, this excess scaling is absorbed into $\boldsymbol{\lambda}$ as: $\boldsymbol{\lambda} = round(1/\gamma_1\gamma_2)$.

In the tensor case, we absorb the scaling of the $\boldsymbol{\lambda}$ output of the real-valued solution into the factor matrices as in "Round", and extend the Scale-and-round matrix approach accordingly.

**AILS: Alternating Integer Least Squares approach:** We used the Integer Least Squares (ILS) with box constraints approach which is proposed in [97, 96]. This approach was recently unified within an Integer Matrix Factorization framework [94]. We exploit the redundancy among ILS problems targeting the same factor matrix, so that the QR factorization in the reduction phase is only computed once. Note that solving general ILS problems is NP-hard [94], which is reflected in the runtime of this method in the experiments. We enabled the extraction of the integer $\boldsymbol{\lambda}$ values through an ILS by noticing that vectorizing the original problem as

$$\min \left\{ ||vec(\boldsymbol{U}\ \boldsymbol{\Lambda}\ \boldsymbol{V}^T) - vec(\boldsymbol{X})||_F^2 \;\middle|\; \boldsymbol{\Lambda} \in \mathbb{Z}_+^{R\times R} \right\}$$

can be transformed to [115]: $\min \left\{ ||(\boldsymbol{V} \odot \boldsymbol{U})\boldsymbol{\lambda} - vec(\boldsymbol{X})||_F^2 \;\middle|\; \boldsymbol{\lambda} \in \mathbb{Z}_+^{R} \right\}$ which gives the ILS to solve for. Note that we attempted to extend this approach for tensor input; however, the materialization of the Khatri-Rao product of all the factor matrices failed due to out of memory problems even for the smallest target rank for both of the datasets used. To illustrate the magnitude of this issue, the size needed for the Khatri-Rao product of all factor matrices for Sutter data and $R = 5$ is: $248347 * 552 * 555 * 5 * 8$ bytes $\approx 3$ Terabytes.

*Evaluation metrics*

We evaluate the methods under comparison in terms of the trade-off between execution time and accuracy for various target ranks considered ($R = \{5, 10, 20, 40\}$). Accuracy is measured in terms of fit: $1 - ||\boldsymbol{X} - \hat{\boldsymbol{X}}||_F / ||\boldsymbol{X}||_F$, where $\hat{\boldsymbol{X}}$ is the re-constructed input through the model factors (this extends trivially to the tensor case); fit can be considered as the the proportion of data explained by the model.

*Initialization details*

In all experiments, when we compare SUSTain and AILS, we provide them with the same initialization.

Regarding the accuracy-time trade-off evaluation, we initialize with several schemes and for each method we choose the one providing the highest fit. The schemes are the following: a) round heuristic, b) scale-and-round heuristic, c) random: random initialization with integers within the required range and $\boldsymbol{\lambda}$ set to all-ones vector, d) random & sampling: random initialization of the patients factor and sampling from the input data to populate the rest of the factors. In the matrix case, we initialize each $j$-th column of $\boldsymbol{V}$ by random sampling of input patient vectors and scaling them to lie on $\mathbb{Z}_\tau$ if needed. In the tensor case, for each sampled slice $\mathcal{X}(i, :, :)$, we populate each $j$-th component of $\boldsymbol{A}^{(2)}, \boldsymbol{A}^{(3)}$ by sampling the row and column of $\mathcal{X}(i, :, :)$ with the maximum sum. Note that when we measure execution time for each approach, we do take into account the time spent for its initialization.

In the scalability evaluation, we initialize each method with the random & sampling scheme (d) described above; this provided better starting points than using pure random initialization. For this experiment, we ignore the initialization time, since we want to focus on the methods' scalability behavior.

*Implementation details*

We used MatlabR2017b for our implementations, along with functionalities for sparse tensors from the "Tensor Toolbox" [114] and for nonnegative matrix factorization from the "nonnegfac-matlab" [12] toolbox. The ILS solver we use for the AILS baseline is included in the state-of-the-art MILES software [116].

The zero-lock problem refers to the case when a single column is zeroed out, thus zeroing out an entire rank-1 component of the solution. To avoid that in our scheme, we add the smallest perturbation possible $(+1)$ to a randomly-chosen coordinate of the vector zeroed out.

In both SUSTain and AILS, we break the iterations when the successive difference of the objective drops below $1e-4$. Finally, the parameter $\tau$ is set to $5$ driven by discussions with medical experts and similarity to many medical scoring systems.

*Hardware*

We conducted our experiments on a server running Ubuntu 14.04 with 1TB of RAM and four Intel E5-4620 v4 CPU's with a maximum clock frequency of 2.10GHz. Each of the processors contains $10$ cores with 2 threads each.

### 3.4.2 Matrix case experiments

**Accuracy-Time trade-off:** In Figure 3.1, we showcase the accuracy-time trade-off regarding the Sutter PAMF dataset. $SUSTain_M$ is at least $60\times$ faster ($R = 40$) than the most accurate baseline (AILS). For $R = 5$, $SUSTain_M$ achieves $425\times$ speedup over AILS: as compared to the $\approx 22$ **minutes** spent by AILS, our approach executes in $\approx$ **3 seconds** for the same level of accuracy. Even for $R = \{10, 20\}$ $SUSTain_M$ achieves $98\times$ and $110\times$ faster computations than AILS. At the same time, $SUSTain_M$ achieves up to $16\%$ higher fit than the scale-and-round heuristic, operating on comparable running times. Note that for $R = 5$, our approach is even faster (and more accurate) than the scale-and-round baseline

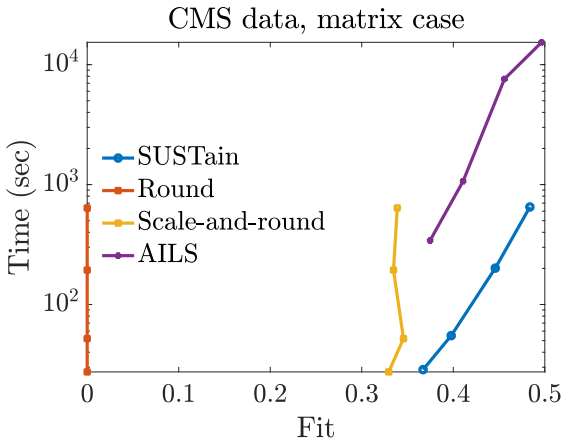



Figure 3.2: Fit (range $[0, 1]$) vs time trade-off for varying target number of phenotypes $R = \{5, 10, 20, 40\}$ for the CMS matrix input. $SUSTain_M$ is at least an order of magnitude faster than the most accurate baseline (up to $38\times$ faster for $R = 20$), while achieving the same level of accuracy. Also, $SUSTain_M$ achieves up to $14\%$ higher fit over scale-and-rounding heuristics.

Table 3.4: Running time (seconds) of one iteration for increasingly larger number of patients considered from the CMS data. Matrix case, $R = 10$.

| **#patients (≈Thousands)** | **246** | **493** | **739** | **985** |
|---|---|---|---|---|
| **#nnz (≈Millions)** | **14** | **27** | **41** | **55** |
| $SUSTain_M$ | 0.71 | 0.95 | 1.66 | 2.82 |
| Round / Scale-and-round | 4.4 | 8.9 | 12.9 | 19.5 |
| AILS | 339 | 514 | 940 | 1254 |

as well, since initializing with random factors provided a better final fit than initializing with the scale-and-round result. We also remark that the naive round heuristic achieves a fit of zero, which is a by-product of zeroing out the majority of the model factor elements.

In Figure 3.2, we provide the results of the same experiment regarding the CMS dataset. For the same level of accuracy, $SUSTain_M$ is at least an order of magnitude faster than AILS, and up to $38\times$ faster for $R = 20$. It also achieves up to $14\%$ higher fit over the scale-and-rounding heuristic for comparable execution time.

**Scaling for larger number of patients:** In Table 3.4, for fixed $R = 10$, we measure a single iteration's time for increasing subsets of CMS patients. The NMF execution time is considered for Round and Scale-and-round heuristics, since their post-processing cost is

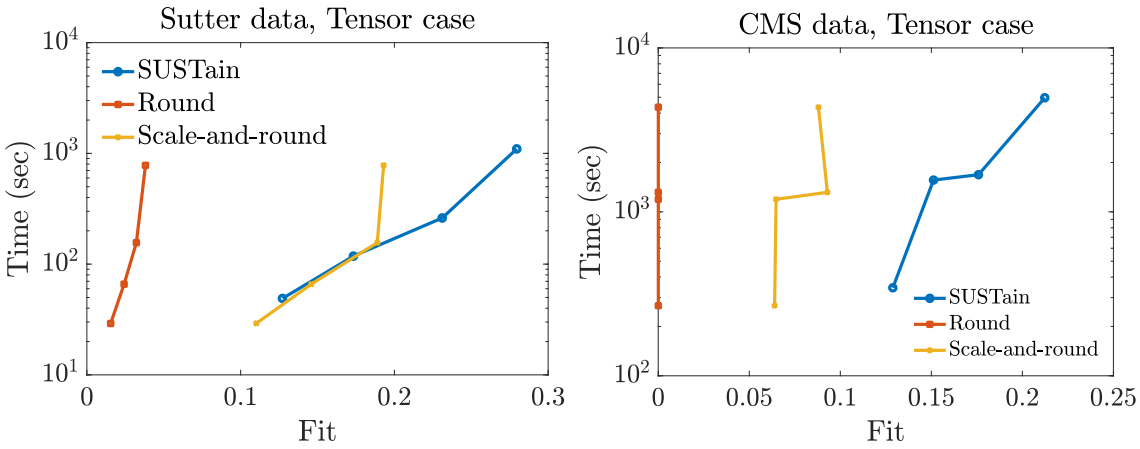



Figure 3.3: Fit (range $[0, 1]$) vs time trade-off for varying target number of phenotypes $R = \{5, 10, 20, 40\}$ for the Sutter and the CMS tensor input. $SUSTain_T$ achieves up to $9\%$ and $12\%$ higher fit respectively over scale-and-rounding heuristics.

Table 3.5: Running time (seconds) of one iteration for increasingly larger number of patients considered from the CMS data. Tensor case, $R = 10$.

| **#patients (≈Thousands)** | **246** | **493** | **739** | **985** |
|---|---|---|---|---|
| **#nnz (≈Millions)** | **29** | **58** | **88** | **117** |
| $SUSTain_T$ | 38.5 | 76.9 | 115 | 151 |
| Round / Scale-and-round | 39.6 | 78 | 117 | 157 |

negligible. $SUSTain_M$ can execute very fast (a single iteration in $\approx 3$ seconds) even for $\approx 985$ thousand patients.

### 3.4.3 Tensor case experiments

**Accuracy-Time trade-off:** In Figure 3.3, we provide the fit-time trade-off for varying target rank of our input tensor datasets. As discussed in Section 3.4.1, the extension of AILS approach to tensors cannot scale for any dataset or target rank considered. Overall, $SUSTain_T$ achieves up to $9\%$ and $12\%$ increase in fit over the scale-and-round heuristic w.r.t. the Sutter PAMF and CMS datasets respectively. Note that the fit of the scale-and-round approach decreases for successively increasing target rank values (e.g., transitioning from $R = 20$ to $R = 40$ for CMS data). This indicates that heuristic approaches which simply post-process real-valued solutions may not fully exploit the available target rank.

**Scaling for larger number of patients:** In Table 3.5, we report the time spent for one iteration of increasingly larger subset of patients considered from the CMS data, with fixed target rank ($R = 10$). The time measured for the heuristic approaches corresponds to the execution time of CP-ALS, since the post-processing cost is negligible. We observe that $SUSTain_T$ achieves linear scale-up w.r.t. increasing number of patients. We also remark that the dominant cost in both $SUSTain_T$ and the CP-ALS is the MTTKRP computation, which explains the comparable running time.

### 3.4.4 Case study on Phenotyping HF patients

Cardiovascular disease (CVD) is the leading cause of death worldwide and heart failure (HF) is a dominant cause of morbidity and mortality. HF is traditionally characterized by reduced ejection fraction (HFrEF) and preserved ejection fraction (HFpEF). But, recent evidence suggests that HF is more heterogeneous than is reflected by ejection fraction. We used SUSTain to explore this heterogeneity in an incident HF cohort.

**Cohort and data selection:** We select only the HF case patients from the Sutter PAMF dataset. For each incident HF case, we extracted data in the 12-months before and the 12-months after the initial HF diagnosis date, which resulted in $70,531$ clinical encounters. We used all the data modalities available, i.e., medication orders and indications and encounter diagnoses. The size of the resulting (patient-by-diagnosis-by-medication) tensor is $3,497 \times 396 \times 367$; the tensor contains a total of $92,662$ non-zero elements.

**Choosing the number of phenotypes:** We use the stability-driven criterion introduced in [117]. The intuition behind this criterion is in promoting a target rank for which several runs with different initial points return reproducible factors. We choose the diagnosis factor matrix as the factor under assessment. Let $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ be the diagnosis factor matrix for 2 different runs with the same target rank. Then, the cross-correlation matrix $\boldsymbol{C} \in \mathbb{R}^{R \times R}$ is computed between the columns of $\boldsymbol{D}_1, \boldsymbol{D}_2$ and the dissimilarity between them is computed

Table 3.6: $SUSTain_T$ achieves $\approx 8.6\%$ increase in fit than a Nonnegative CP-ALS model truncated to achieve the same level of sparsity. The result refers to the HF case study for $R = 15$.

| **method** | $\#\mathbf{nnz}(\mathbf{A^{(1)}})$ | $\#\mathbf{nnz}(\mathbf{A^{(2)}})$ | $\#\mathbf{nnz}(\mathbf{A^{(3)}})$ | **fit** |
|---|---|---|---|---|
| $SUSTain_T$ | $3,438$ | $54$ | $88$ | $0.261$ |
| NN CP-ALS | $3,497$ | $60$ | $90$ | $0.175$ |

as [117]:

$$diss(\boldsymbol{D}_1, \boldsymbol{D}_2) = \frac{1}{2R}\left(2R - \sum_{j=1}^{R} max_{1\leq k\leq R}\boldsymbol{C}(k,j) - \sum_{k=1}^{R} max_{1\leq j\leq R}\boldsymbol{C}(k,j)\right)$$

Note that when $\boldsymbol{D}_1$ can be transformed to $\boldsymbol{D}_2$ by column permutation, then $diss(\boldsymbol{D}_1, \boldsymbol{D}_2) = 0$. If $B$ is the number of repetitions for each target rank, then the following relation computes the average dissimilarity over $B(B-1)/2$ pairs of resulting factors:

$$Y(R) = \frac{2}{B(B-1)} \sum_{1\leq b<b'\leq B} diss(\boldsymbol{D}_b, \boldsymbol{D}_{b'})$$

We used the "staNMF" toolbox [3] to compute the above score for each target rank on the range $\{5, \ldots, 20\}$. The input to $SUSTain_T$ were $B = 20$ initial points of the round heuristic. $R = 15$ phenotypes were selected based on the above criterion. For the target rank chosen, we pick the solution yielding the highest fit.

**SUSTain provides concise and accurate solutions:** We observed that besides preserving the input data properties and providing a natural interpretation for medical experts, SUSTain implicitly imposes sparse factors. To assess the factors' conciseness, we compare their fit with the achieved fit of the real-valued model (NN CP-ALS), which is post-processed to achieve factor sparsity (as would be done by a practitioner). For each of the feature factors (diagnosis and medication) of the real-valued model, we only consider the top-$k$ elements for each column (i.e., most important elements of each phenotype). For the patient factor, we consider the top-$k$ elements for each row (i.e., most important phenotypes for each pa-

[3] `https://github.com/bdgp/staNMF`

Table 3.7: Representative phenotypes extracted by $SUSTain_T$ for our HF case study. The score of each feature indicates its relative frequency within the phenotype. The prefix for each feature indicates whether it corresponds to a medication (Rx) or a diagnosis (Dx). A cardiologist provided phenotype annotations and validated that: the top-most phenotype is aligned to guideline-based management of HF with reduced LVEF (HFrEF), the next one corresponds to typical hypertensive patients (common risk factor of HF) and the last one corresponds to hypertensive patients being more difficult to control.

| **HF with reduced LVEF (HFrEF)** | **Score** |
|---|---|
| Rx_Loop Diuretics | 3 |
| Dx_Congestive heart failure | 1 |
| Rx_ACE Inhibitors | 1 |
| Rx_Alpha-Beta Blockers | 1 |
| Rx_Potassium | 1 |
| **Hypertension** | **Score** |
| Rx_ACE Inhibitors | 3 |
| Dx_Essential hypertension | 1 |
| Rx_Alpha-Beta Blockers | 1 |
| Rx_Beta Blockers Cardio-Selective | 1 |
| Rx_Calcium Channel Blockers | 1 |
| Rx_HMG CoA Reductase Inhibitors | 1 |
| Rx_Loop Diuretics | 1 |
| Rx_Thiazides and Thiazide-Like Diuretics | 1 |
| **Hypertension (more difficult to control)** | **Score** |
| Rx_Angiotensin II Receptor Antagonists | 2 |
| Rx_Beta Blockers Cardio-Selective | 2 |
| Rx_Calcium Channel Blockers | 2 |
| Dx_Essential hypertension | 1 |
| Rx_Antiadrenergic Antihypertensives | 1 |
| Rx_Loop Diuretics | 1 |
| Rx_Potassium | 1 |

tient). In each case, the value of $k$ is chosen so that the sparsity level is close to the one achieved by $SUSTain_T$. We provide the results in Table 3.6, where we notice that for the same level of sparsity, $SUSTain_T$ achieves $\approx 8.6\%$ increase in fit. Thus, the integer factors of $SUSTain_T$ decompose the input more accurately for the same level of sparsity than the real-valued counterpart.

**Phenotype discovery:** In Table 3.1 and Table 3.7, we provide representative phenotypes extracted through our method. A subset of annotations provided by the cardiologist are as follows: hyperlipidemia (the one in Table 3.1), HF with reduced LVEF (HFrEF), hypertension (HTN), HTN which is more difficult to control, persistent and chronic atrial fibrilla-

tion, depression, diabetes, comorbidities of aging, prior pulmonary embolism. Overall, 13 out of 15 phenotype candidates were annotated as clinically meaningful phenotypes related to heart failure.

### 3.5 Related Work

**Discrete factorization-based approaches:** Dong et al. [94] proposed an Integer Matrix Factorization framework via solving Integer Least Squares subproblems. As we experimentally evaluated, this approach is orders of magnitude slower than SUSTain while achieving the same level of accuracy. Kolda and O'Leary [118, 104] proposed a Semidiscrete Matrix Decomposition into factors containing ternary values ($\{-1, 0, 1\}$). Despite its demonstrated success for compression purposes, a direct application of this approach would introduce negative values into the factors, thus hurting interpretability for nonnegative input. Finally, several prior works target binary factorization (e.g., [119, 120, 121, 122, 123, 124, 125]). In contrast to strictly binary factors, SUSTain captures the quantity embedded in the input data, which reveals important information (e.g., relative phenotype prevalence and associated feature frequencies).

**Unsupervised Phenotyping:** Extensive prior work applies factorization techniques for unsupervised phenotyping (e.g., [24, 21, 49, 50, 51, 126]). However, no work considered extracting scoring-based phenotypes to facilitate their interpretation by domain experts.

**HALS fitting algorithms:** Our fitting algorithms follow the intuition of Hierarchical Alternating Least Squares (HALS) framework [127] (aka rank-one residue iteration [128]), which enables formulating the solution for each $k$-th rank-1 component separately. However, plain HALS does not tackle the challenges involved with either imposing integer constraints or solving for the vector $\boldsymbol{\lambda}$.

## 3.6 Conclusions

The accuracy and scalability of SUSTain on "native" integer data derives from two key insights. One is expected: just rounding or applying related transformations to real-valued solutions is inherently limited. The second may be more surprising: while discrete constraints might appear to make the problem more challenging, in fact, a careful organization of the problem into subparts can mitigate that complexity. In our case, we identify a problem partitioning of integer-constrained subproblems that leads to an optimal and efficient solution; and, we also define the order of alternating updates so as to enable reuse of shared intermediate results. Consequently, SUSTain outperforms several baselines on both synthetic (publicly-available) and real EHR data, showing either a better fit or orders-of-magnitude speedups at a comparable fit.

Moving forward, there are many other sources of integer values in real-world data. These include, for instance, ordinal values. Thus, whereas this work targets event counts, extensions for other cases is a ripe target for future work.

Lastly, to enable reproducibility of our work, we open-source our implementations and make them publicly available at: `https://github.com/kperros/SUSTain`.

# CHAPTER 4

# SPARTAN: SCALABLE PARAFAC2 FOR LARGE & SPARSE DATA


Summary

Aggregating event occurrences over time for unsupervised phenotyping purposes, as done in Chapter 3, may not be appropriate when disease states are evolving, as it obfuscates the temporal ordering and structure of individual EHR encounter records. In this Chapter, we expose how PARAFAC2 can account for the folded structure of EHRs and augment phenotype candidates with information about their timing and sequencing across different patients, which may in turn lead to a better understanding of disease progression. To enable scalable PARAFAC2 computations for large EHRs, we propose SPARTan, which exploits the inherent sparsity of input EHR data and shows both faster (up to $22\times$) and more memory-efficient performance than prior work.


A common problem in unsupervised learning is how to extract the latent correlation structure among a set of variables, measured across a set of subjects whose observations do not align naturally. For example, when modeling features derived from EHRs across a set of patients, the number and duration of treatments may vary widely in time, meaning there is no meaningful way to align their clinical records across time points. This renders the extraction of high-order relationships among observations, patients and features a challenging task, e.g., extracting correlated sets of features indicating clinical manifestations, the corresponding subset of patients exhibiting those features, as well as when these are exhibited across observations for different patients. Extracting those high-order relationships from large-scale EHR data may offer a scalable way to better understand the progression of diseases and disease subtypes. We expose how the PARAFAC2 model can be utilized

Table 4.1: Running time comparison: Time in minutes of one iteration for increasingly larger datasets (63m to 500m) and fixed target rank (two cases considered: $R = \{10, 40\}$). The mode sizes for the datasets constructed are: 1Mil. subjects, 5K variables and a maximum of 100 observations per subject. **OoM** (Out of Memory) denotes that the execution failed due to the excessive amount of memory requested. Experiments are conducted on a server with 1TB of RAM.

| **Target Rank** | **10** | | | | **40** | | | |
|---|---|---|---|---|---|---|---|---|
| **#nnz(Millions)** | **63** | **125** | **250** | **500** | **63** | **125** | **250** | **500** |
| SPARTan | 7.4 | 8.9 | 11.5 | 15.4 | 14 | 18.4 | 61 | 114 |
| Sparse PARAFAC2 | 24.4 | 60.1 | 72.3 | 194.5 | 275.2 | 408.1 | **OoM** | **OoM** |

to tackle this challenge, yielding interpretable and robust output. However, the standard algorithm fitting the PARAFAC2 model expects dense data as input; designing an efficient algorithm which can exploit the inherent sparsity of EHR data and can handle large-scale EHR datasets has been an open challenge.

In this work, we develop a scalable method to compute the PARAFAC2 model for large and sparse input data. Our method, called SPARTan, exploits special structure of intermediate data arising within the PARAFAC2 model fitting, leading to a novel algorithm that is both faster and more memory-efficient than prior work. We evaluate SPARTan on both synthetic and real datasets, showing $22\times$ performance gains over the best previous implementation and also handling larger problem instances for which the baseline fails.

## 4.1 Introduction

This work concerns tensor-based analysis and mining of multi-modal data where observations are difficult or impossible to align naturally along one of its modes. A concrete example of such data is electronic health records (EHR), our primary motivating application. An EHR dataset contains longitudinal patient information, represented as an event sequence of multiple modalities such as diagnoses, medications, procedures, and lab results. An important characteristic of such event sequences is that there is no simple way to align observations in time across patients. For instance, different patients may have varying length records between the first admission and the most recent hospital discharge; or, two

patients whose records' have the same length may still not have a sensible chronological alignment as disease stages and patient progress vary.

For tensor methods, such data poses a significant challenge. Consider the most popular tensor analysis method in data mining, the canonical polyadic (CP) factorization (also known as PARAFAC or CANDECOMP) [23, 89, 108]. A dataset with three modes might be stored as an $I \times J \times K$ tensor $\mathcal{X}$, which CP then decomposes into a sum of multi-way outer (rank-one) products, $\mathcal{X} \approx \sum_{r=1}^{R} \boldsymbol{u}_r \circ \boldsymbol{v}_r \circ \boldsymbol{w}_r$, where $\boldsymbol{u}_r, \boldsymbol{v}_r, \boldsymbol{w}_r$ are column vectors of size $I, J, K$, respectively, that effectively represents latent data concepts. Its popularity owes to its *intuitive output structure* and *uniqueness* property that makes the model reliable to interpret [129, 130, 11, 131, 132], as well as the existence of scalable algorithms and software [102, 133, 114]. In the context of our main motivating application, tensor methods, such as CP, may offer a scalable way to better understand disease and disease subtype progression; this may be achieved through the extraction of disease and treatment patterns, the corresponding subsets of patients exhibiting those patterns, as well as their temporal trends indicating when each of those patterns is expressed across observations for different patients. However, in order to extract the latent correlation structure among variables, patients and time from EHR data, it would require finding some way to align time. This fact is an inherent limitation of applying the CP model: any preprocessing to aggregate across time may lose temporal patterns [24, 134, 135], while more sophisticated temporal feature extraction methods typically need continuous and sufficiently long temporal measures to work [136]. Other proposed methods specific to healthcare applications may give some good results [137, 138, 139] but lack the *uniqueness* guarantee; thus, it becomes harder to reliably extract the actual latent concepts as an equivalent arbitrary rotation of them will provide the same fit. All of these weaknesses apply in the EHR scenario outlined above.

In fact, the type of data in the motivating example are quite general: consider that we have $K$ subjects, for which we record $J$ variables and we permit each $k$-th subject to have $I_k$ observations, which are not necessarily comparable among the different subjects. For this

type of data, Harshman proposed the *PARAFAC2 model* [58]. It is a more flexible version of CP: while CP applies the same factors across a collection of matrices, PARAFAC2 instead applies the same factor along one mode and allows the other factor matrix to vary [11]. At the same time, it preserves the desirable properties of CP, such as uniqueness [140, 141, 59, 142]. As shown in Figure 4.1, PARAFAC2 approximates each one of the input matrices as: $\boldsymbol{X}_k \approx \boldsymbol{U}_k\ \boldsymbol{S}_k\ \boldsymbol{V}^T$, where $\boldsymbol{U}_k$ is of size $I_k \times R$, $\boldsymbol{S}_k$ is a diagonal $R$-by-$R$, $\boldsymbol{V}$ is of size $J \times R$ and $R$ is the target rank of the decomposition.

Despite its applicability, the lack of efficient PARAFAC2 decomposition algorithms has been cited as a reason for its limited popularity [143, 144]. Overall, PARAFAC2 has been mostly used for dense data (e.g., [59]) or sparse data with a small number of subjects [144]. To our knowledge, no work has assessed PARAFAC2 for large-scale sparse data, as well as the challenges arising by doing so.

In this work, we propose SPARTan (abbreviated from Scalable PARafac Two) to fill this gap, with a focus on achieving scalability on large and sparse datasets. Our methodological advance is a new algorithm for scaling up the core computational kernel arising in the PARAFAC2 fitting algorithm. SPARTan achieves the best of both worlds in terms of speed and memory efficiency: *a)* it is *faster* than a highly-optimized baseline in all cases considered for both real (Figures 4.5, 4.6, 4.7) and synthetic (Table 4.1) datasets, achieving up to $22\times$ performance gain; *b)* at the same time, SPARTan is *more scalable*, in that it can execute in reasonable time for large problem instances when the baseline fails due to excessive memory consumption (Table 4.1). We summarize our contributions as:

- **Scalable PARAFAC2 method:** We propose SPARTan, a scalable algorithm fitting the

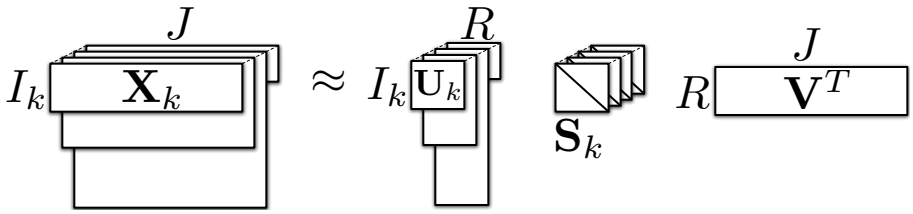



Figure 4.1: Illustration of the PARAFAC2 model.

PARAFAC2 model on large and sparse data.

- **Evaluation on various datasets:** We evaluate the scalability of our approach using datasets originating from two different application domains, namely a longitudinal EHR and a time-evolving movie ratings' dataset, which is also publicly available. Additionally, we perform synthetic data experiments.

- **Real-world case study**: We performed a case study of applying SPARTan on temporal phenotyping over medically complex pediatric patients in collaboration with Children's Healthcare of Atlanta (CHOA). The phenotypes and temporal trends discovered were endorsed by a clinical expert from CHOA.

To promote reproducibility, our code is open-sourced and publicly available at: `https://github.com/kperros/SPARTan`.

## 4.2 Background

Next we describe the necessary terminology and operations regarding tensors. Then, we provide an overview of the CP model and relevant fitting algorithm. In Table 4.2, we summarize the notations used throughout this work.

Table 4.2: Notations used throughout this work.

| Symbol | Definition |
|---|---|
| $\mathcal{X}, \boldsymbol{X}, \boldsymbol{x}, x$ | Tensor, matrix, vector, scalar |
| $\boldsymbol{X}^{\dagger}$ | Moore-Penrose pseudoinverse |
| $\boldsymbol{X}(:, i)$ | Spans the entire $i$-th column of $\boldsymbol{X}$ (same for tensors) |
| $\boldsymbol{X}(i, :)$ | Spans the entire $i$-th row of $\boldsymbol{X}$ (same for tensors) |
| $diag(\boldsymbol{x})$ | Diagonal matrix with vector $\boldsymbol{x}$ on the diagonal |
| $diag(\boldsymbol{X})$ | Extract diagonal of matrix $\boldsymbol{X}$ |
| $\boldsymbol{X}_k$ | shorthand for $\boldsymbol{X}(:, :, k)$ (k-th frontal slice of tensor $\mathcal{X}$) |
| $\{\boldsymbol{X}_k\}$ | the collection of $\boldsymbol{X}_k$ matrices, for all valid $k$ |
| $\boldsymbol{X}_{(n)}$ | mode-$n$ matricization of tensor $\mathcal{X}$ |
| $\circ$ | Outer product |
| $\otimes$ | Kronecker product |
| $\odot$ | Khatri-Rao product |
| $*$ | Hadamard (element-wise) product |

### 4.2.1 Tensors and Tensor Operations

The *order* of a tensor denotes the number of its dimensions, also known as ways or modes (e.g., matrices are 2-order tensors). A *fiber* is a vector extracted from a tensor by fixing all modes but one. For example, a matrix column is a mode-1 fiber. A *slice* is a matrix extracted from a tensor by fixing all modes but two. In particular, the $\boldsymbol{X}(:,:,k)$ slices of a third-order tensor $\mathcal{X}$ are called the frontal ones and we succinctly denote them as $\boldsymbol{X}_k$ [11]. *Matricization*, also called *reshaping* or *unfolding*, logically reorganizes tensors into other forms without changing the values themselves. The mode-$n$ matricization of a $N$-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is denoted by $\boldsymbol{X}_{(n)} \in \mathbb{R}^{I_n \times I_1 I_2 \ldots I_{n-1} I_{n+1} \ldots I_N}$ and arranges the mode-$n$ fibers of the tensor as columns of the resulting matrix.

### 4.2.2 CP Decomposition

The CP decomposition [23, 89, 108] of a third-order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ is its approximation by a sum of three-way outer products:

$$\mathcal{X} \approx \sum_{r=1}^{R} \boldsymbol{u}_r \circ \boldsymbol{v}_r \circ \boldsymbol{w}_r \tag{4.1}$$

where $\boldsymbol{u}_r \in \mathbb{R}^I, \boldsymbol{v}_r \in \mathbb{R}^J$ and $\boldsymbol{w}_r \in \mathbb{R}^K$ are column vectors. If we assemble the column vectors $\boldsymbol{u}_r, \boldsymbol{v}_r, \boldsymbol{w}_r$ as: $\boldsymbol{U} = [\boldsymbol{u}_1\ \boldsymbol{u}_2\ \ldots\ \boldsymbol{u}_R] \in \mathbb{R}^{I \times R}, \boldsymbol{V} = [\boldsymbol{v}_1\ \boldsymbol{v}_2\ \ldots\ \boldsymbol{v}_R] \in \mathbb{R}^{J \times R}, \boldsymbol{W} = [\boldsymbol{w}_1\ \boldsymbol{w}_2\ \ldots\ \boldsymbol{w}_R] \in \mathbb{R}^{K \times R}$, then $\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}$ are called the *factor matrices*. Interpretation of CP is very intuitive: we consider that the input tensor can be summarized as $R$ latent concepts. Then, for each $r$-th concept, the vectors $(\boldsymbol{u}_r, \boldsymbol{v}_r, \boldsymbol{w}_r)$ are considered as soft-clustering membership indicators, for the corresponding I, J and K elements of each mode. An equivalent formulation of Relation (4.1) w.r.t. the frontal slices $\boldsymbol{X}_k$ of the input tensor $\mathcal{X}$ is [145]:

$$\boldsymbol{X}_k \approx \boldsymbol{U}\ \boldsymbol{S}_k\ \boldsymbol{V}^T \tag{4.2}$$

where $k = 1, 2, \ldots, K$ and $\mathcal{S} \in \mathbb{R}^{R \times R \times K}$ is an auxiliary tensor. Each frontal slice $\boldsymbol{S}_k$ of $\mathcal{S}$ contains the row vector $\boldsymbol{W}(k,:)$ along its diagonal: $\boldsymbol{S}_k = diag(\boldsymbol{W}(k,:))$. Relation (4.2) provides another viewpoint of interpreting the CP model, through its correspondence to the Singular Value Decomposition (SVD): each slice $\boldsymbol{X}_k$ is decomposed to a set of factor matrices $\boldsymbol{U}, \boldsymbol{V}$ (similar to the singular vectors) which are common for all the slices, and a diagonal middle matrix (similar to the singular values) which varies for each $k$-th slice. Note, however, that no orthogonality constraints are imposed on $\boldsymbol{U}, \boldsymbol{V}$ of the CP model, as in the SVD [144].

**Uniqueness.** A fundamental property of CP is uniqueness [129, 130]. The issue with non-uniqueness can be exemplified via matrix factorization as follows [11, 131]: If a matrix $\boldsymbol{X}$ is approximated by the product of $\boldsymbol{A}\boldsymbol{B}^T$, then it can also be approximated with the same error by $\boldsymbol{A}\boldsymbol{Q}\boldsymbol{Q}^{-1}\boldsymbol{B}^T = \tilde{\boldsymbol{A}}\tilde{\boldsymbol{B}}^{\boldsymbol{T}}$, for any invertible $\boldsymbol{Q}$. Thus, we can easily construct two completely different sets of rank-one factors that sum to the original matrix. Inevitably, this hurts interpretability, since we cannot know whether our solution is an arbitrarily rotated version of the actual latent factors. In contrast to matrix factorization or Tucker decomposition [11], Kruskal [129] proved that CP is unique, under the condition: $k_{\boldsymbol{U}} + k_{\boldsymbol{V}} + k_{\boldsymbol{W}} \geq 2R + 2$, where $k_{\boldsymbol{U}}$ is the $k$-rank of $\boldsymbol{U}$, defined as the maximum value $k$ such that any $k$ columns are linearly independent. The only exception is related to elementary indeterminacies of scaling and permutation of the component vectors [11, 132]. In sum, the CP decomposition is pursuing the true underlying latent information of the input tensor and provides reliable interpretation for unsupervised approaches.

**Fitting the CP model.** Perhaps the most popular algorithm for fitting the CP model is the CP-Alternating Least Squares (CP-ALS) [108, 89], listed in Algorithm 4. The main idea is to solve for one factor matrix at a time, by fixing the others. In that way, each subproblem is reduced to a linear least-squares problem. In case the input tensor contains non-negative values, a non-negative least-squares solver (e.g., [146]) can be used instead of an unconstrained one, to further improve the factors' interpretability [143].

Due to the ever increasing need for CP decompositions in data mining, the parallel CP-ALS for sparse tensors has been extensively studied in the recent literature for both single-node and distributed settings (e.g., [102, 147, 148, 149, 150, 151]). A pioneering work in addressing scalability issues for sparse tensors was provided by Bader and Kolda [102] [1]. The authors identified and scaled up the algorithm's bottleneck, which is the materialization of the Matricized-Tensor-Times-Khatri-Rao-Product (MTTKRP). For example, in Algorithm 4, the MTTKRP corresponds to the computation of $\boldsymbol{X}_{(1)}(\boldsymbol{W} \odot \boldsymbol{V})$ when solving for $\boldsymbol{U}$. For large and sparse tensors, a naive construction of the MTTKRP requires huge memory (which might not even be available) and computational cost and has to be avoided.

**Algorithm 4** CP-ALS

**Input:** $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ and target rank $R$
**Output:** $\boldsymbol{\lambda} \in \mathbb{R}^{R}, \boldsymbol{U} \in \mathbb{R}^{I \times R}, \boldsymbol{V} \in \mathbb{R}^{J \times R}, \boldsymbol{W} \in \mathbb{R}^{K \times R}$
1: Initialize $\boldsymbol{V}, \boldsymbol{W}$
2: **while** convergence criterion is not met **do**
3: $\quad \boldsymbol{U} \leftarrow \boldsymbol{X}_{(1)}(\boldsymbol{W} \odot \boldsymbol{V})(\boldsymbol{W}^T\boldsymbol{W} * \boldsymbol{V}^T\boldsymbol{V})^{\dagger}$
4: $\quad$ Normalize columns of $\boldsymbol{U}$
5: $\quad \boldsymbol{V} \leftarrow \boldsymbol{X}_{(2)}(\boldsymbol{W} \odot \boldsymbol{U})(\boldsymbol{W}^T\boldsymbol{W} * \boldsymbol{U}^T\boldsymbol{U})^{\dagger}$
6: $\quad$ Normalize columns of $\boldsymbol{V}$
7: $\quad \boldsymbol{W} \leftarrow \boldsymbol{X}_{(3)}(\boldsymbol{V} \odot \boldsymbol{U})(\boldsymbol{V}^T\boldsymbol{V} * \boldsymbol{U}^T\boldsymbol{U})^{\dagger}$
8: $\quad$ Normalize columns of $\boldsymbol{W}$ and store norm in $\boldsymbol{\lambda}$
9: **end while**

## 4.3 PARAFAC2 Overview & Challenges

### 4.3.1 Model

As we introduced in Section 4.1, the PARAFAC2 model [58] can successfully deal with an incomparable mode of each slice $\boldsymbol{X}_k$ [59]. It does so, by introducing a set of $\boldsymbol{U}_k$ matrices replacing the $\boldsymbol{U}$ matrix of the CP model in Relation (4.2). Thus, each slice $\boldsymbol{X}_k$ is

[1] The contributions of [102], among others, are summarized as the Tensor Toolbox [114], which is widely acclaimed as the state-of-the-art package for single-node sparse tensor operations and algorithms.

decomposed as shown in Figure 4.1:

$$\boldsymbol{X}_k \approx \boldsymbol{U}_k \, \boldsymbol{S}_k \, \boldsymbol{V}^T \tag{4.3}$$

where $k = 1, \ldots, K, \boldsymbol{U}_k \in \mathbb{R}^{I_k \times R}, \boldsymbol{S}_k \in \mathbb{R}^{R \times R}$ is diagonal and $\boldsymbol{V} \in \mathbb{R}^{J \times R}$. To preserve uniqueness, Harshman [58] imposed the constraint that the cross product $\boldsymbol{U}_k^T \boldsymbol{U}_k$ is invariant regardless which subject $k$ is involved [11, 144]. In that way, the CP model's invariance of the factor $\boldsymbol{U}_k$ itself (or $\boldsymbol{U}$ given its invariance to k), is relaxed [152]. For the above constraint to hold, each $\boldsymbol{U}_k$ factor is decomposed as:

$$\boldsymbol{U}_k = \boldsymbol{Q}_k H \tag{4.4}$$

where $\boldsymbol{Q}_k$ is of size $I_k \times R$ and has orthonormal columns, and $\boldsymbol{H}$ is an $R \times R$ matrix, which does not vary by $k$ [11]. Then, the constraint that $\boldsymbol{U}_k^T \boldsymbol{U}_k$ is constant over $k$ is implicitly enforced, as follows: $\boldsymbol{U}_k^T \boldsymbol{U}_k = \boldsymbol{H}^T \boldsymbol{Q}_k^T \boldsymbol{Q}_k \boldsymbol{H} = \boldsymbol{H}^T \boldsymbol{H} = \boldsymbol{\Phi}$.

There have been several results regarding the uniqueness property of PARAFAC2 [140, 141, 59]. The most relevant [140] towards our large-scale data scenario (i.e., the number of $K$ subjects can easily reach the order of hundreds of thousands) is that a rank-$R$ PARAFAC2 model is unique if $\boldsymbol{\Phi}$ and $\boldsymbol{V}$ have rank $R$, $\boldsymbol{\Phi}$ has no zero entries and the number of $K$ subjects is at least: $R\,(R+1)\,(R+2)\,(R+3)\,/\,24$ [153]. Note that this bound on the number of $K$ subjects is sufficient but not necessary to achieve uniqueness; it is conjectured (as evaluated through simulation studies) that PARAFAC2 provides unique solutions as long as $K \geq 4$ [59, 153, 142].

### 4.3.2 Classical Algorithm for PARAFAC2

Below, we overview the classical algorithm for fitting PARAFAC2, proposed by [59]. Their original algorithm expects dense data as input. Its objective function is as follows:

$$\min_{\{\boldsymbol{U}_k\},\{\boldsymbol{S}_k\},\boldsymbol{V}} \sum_{k=1}^{K} ||\boldsymbol{X}_k - \boldsymbol{U}_k\boldsymbol{S}_k\boldsymbol{V}^T||_F^2$$

subject to: $\boldsymbol{U}_k = \boldsymbol{Q}_k\boldsymbol{H}$, $\boldsymbol{Q}_k^T\boldsymbol{Q}_k = \boldsymbol{I}$ and $\boldsymbol{S}_k$ to be diagonal. Algorithm 5 follows an Alternating Least Squares (ALS) approach, divided in two distinct steps: first (lines 3-6), the set of column-orthonormal matrices $\{\boldsymbol{Q}_k\}$ is computed by fixing $\boldsymbol{H}, \boldsymbol{V}, \{\boldsymbol{S}_k\}$. This step can be derived by examining the pursuit of each $\boldsymbol{Q}_k$ as an individual Orthogonal Procrustes Problem [109] of the form:

$$\min_{\boldsymbol{Q}_k} ||\boldsymbol{X}_k - \boldsymbol{Q}_k\boldsymbol{H}\boldsymbol{S}_k\boldsymbol{V}^T||_F^2 \tag{4.5}$$

subject to $\boldsymbol{Q}_k^T\boldsymbol{Q}_k = \boldsymbol{I}$. Given the SVD of $\boldsymbol{H}\boldsymbol{S}_k\boldsymbol{V}^T\boldsymbol{X}_k^T$ as $\boldsymbol{P}_k\boldsymbol{\Sigma}_k\boldsymbol{Z}_k$, the minimum of objective (4.5) over column-orthonormal $\boldsymbol{Q}_k$ is given by $\boldsymbol{Q}_k = \boldsymbol{Z}_k\boldsymbol{P}_k^T$ [59, 109].

Second, after solving for and fixing $\{\boldsymbol{Q}_k\}$, we find a solution for the rest of the factors as:

$$\min_{\boldsymbol{H},\{\boldsymbol{S}_k\},\boldsymbol{V}} \sum_{k=1}^{K} ||\boldsymbol{Q}_k^T\boldsymbol{X}_k - \boldsymbol{H}\boldsymbol{S}_k\boldsymbol{V}^T||_F^2 \tag{4.6}$$

where $\boldsymbol{S}_k$ is diagonal. Note the equivalence of the above objective with the CP "slice-wise" formulation of Relation (4.2). This equivalence implies that minimizing the objective (4.6) is achieved by executing the CP decomposition on a tensor $\mathcal{Y} \in \mathbb{R}^{R\times J\times K}$ with frontal slices $\boldsymbol{Y}_k = \boldsymbol{Q}_k^T\boldsymbol{X}_k$ (lines 7-10). In order to avoid executing all the costly CP iterations, Kiers et al. [59] propose to run a single CP-ALS iteration, since this suffices to decrease the objective.

The PARAFAC2 model can be extended so that non-negative constraints are imposed on $\{\boldsymbol{S}_k\}, \boldsymbol{V}$ factors [145]. This is a property inherited by the CP-ALS iteration, where we

can constrain the factors $\boldsymbol{V}$ and $\boldsymbol{W}$ to be non-negative, as discussed in Section 4.2.2. Note that constraining the $\{\boldsymbol{U}_k\}$ factors to be non-negative as well is not as simple, and a naive approach would violate the model properties [154].

**Algorithm 5** PARAFAC2-ALS [59]

**Input:** $\{\boldsymbol{X}_k \in \mathbb{R}^{I_k \times J}\}$ for $k = 1, \ldots, K$ and target rank $R$
**Output:** $\{\boldsymbol{U}_k \in \mathbb{R}^{I_k \times R}\}, \{\boldsymbol{S}_k \in \mathbb{R}^{R \times R}\}$ for $k = 1, \ldots, K,\ \boldsymbol{V} \in \mathbb{R}^{J \times R}$
1: Initialize $\boldsymbol{H} \in \mathbb{R}^{R \times R}, \boldsymbol{V},\ \{\boldsymbol{S}_k\}$ for $k = 1, \ldots, K$
2: **while** convergence criterion is not met **do**
3:  **for** $k = 1, \ldots, K$ **do**
4:  $[\boldsymbol{P}_k, \boldsymbol{\Sigma}_k, \boldsymbol{Z}_k] \leftarrow$ truncated SVD of $\boldsymbol{H}\boldsymbol{S}_k\boldsymbol{V}^T\boldsymbol{X}_k^T$ at rank $R$
5:  $\boldsymbol{Q}_k \leftarrow \boldsymbol{Z}_k\boldsymbol{P}_k^T$
6:  **end for**
7:  **for** k=1, …, K **do**
8:  $\boldsymbol{Y}_k \leftarrow \boldsymbol{Q}_k^T\boldsymbol{X}_k$ *// construct slice $\mathbf{Y}_k$ of tensor $\mathcal{Y}$*
9:  **end for**
10:  Run a single iteration of CP-ALS on $\mathcal{Y}$ to compute $\boldsymbol{H}, \boldsymbol{V}, \boldsymbol{W}$
11:  **for** $k = 1, \ldots, K$ **do**
12:  $\boldsymbol{S}_k \leftarrow diag(\boldsymbol{W}(k, :))$
13:  **end for**
14: **end while**
15: **for** $k = 1, \ldots, K$ **do**
16:  $\boldsymbol{U}_k \leftarrow \boldsymbol{Q}_k\boldsymbol{H}$ *// Assemble $\boldsymbol{U}_k$*
17: **end for**

### 4.3.3 Challenges of PARAFAC2 on sparse data

Next, we summarize a set of crucial observations regarding the computational challenges, when executing Algorithm 5 on large, sparse data:

**Bottleneck of Algorithm 5.** Regarding the 1st step (lines 3-6), in practice for sparse $\boldsymbol{X}_k$, each one of the $K$ sub-problems scales as $\mathcal{O}(min(RI^2, R^2I))$, due to the SVD involved [155], where $I$ is an upper bound for $I_k$. Note that this computation can be trivially parallelized for all $K$ subjects. On the other hand, the 2nd step (lines 7-10) is dominated by the MTTKRP computation (which as we discussed in Section 4.2 is the bottleneck of sparse CP-ALS). Thus, it scales as $3R\ nnz(\mathcal{Y})$ [150], using state-of-the-art sparse tensor libraries for single-node [102]. Given that none of the input matrices $\boldsymbol{X}_k$ is completely zero, then: $3R\ nnz(\mathcal{Y}) \geq 3KR^2$. As a result, this step becomes the bottleneck of Algorithm 5 for large and sparse "irregular" tensors, since it cannot be parallelized w.r.t. the $K$ subjects,

as trivially happens with the 1st one.

**Imbalance of mode sizes of $\mathcal{Y}$.** The size of the intermediate tensor $\mathcal{Y}$ formed is $R \times J \times K$. For large-scale data, we expect that $R << K, J$ since $R$ corresponds to the target rank of the overall PARAFAC2 decomposition. Note that this property of "size imbalance" may not hold for a general large tensor, thus generic CP-ALS solvers (e.g., [102]) cannot exploit it.

**Structured sparsity of $\{\boldsymbol{X}_k\}$.** First, we observe that even if the input slices $\{\boldsymbol{X}_k \in \mathbb{R}^{I_k \times J}\}$ are very sparse, all their $I_k$ rows will contain at least one non-zero element. If this is not the case, we can simply filter the zero ones, without affecting the result. However, this does not hold for the $J$ columns of each $\boldsymbol{X}_k$, which have to be aligned across all $K$ subjects. A direct consequence of that in real datasets is that very few variables (relative to their total number) are typically recorded for each subject. For example, in the EHR data use case introduced in Section 4.1, very few medical features are recorded for each patient.

Driven by this observation, we are motivated to computationally exploit any column sparsity (i.e., cases where many columns will be completely zero) of each one of the input matrices $\boldsymbol{X}_k$.

## 4.4 The SPARTan approach

### 4.4.1 Overview

Motivated by the challenges presented in Section 4.3, we propose a specialized version of the Matricized-Tensor-Times-Khatri-Rao-Product (MTTKRP) kernel, specifically targeting the intermediate tensor $\mathcal{Y} \in \mathbb{R}^{R \times J \times K}$ formed within the PARAFAC2-ALS algorithm. We first provide an overview of the properties that our approach exhibits:

- It is *fully parallelizable w.r.t. the $K$ subjects*. This property is crucial towards scaling up for large-scale "irregular" tensors.
- It *exploits the structured sparsity* of the input frontal slices $\{\boldsymbol{X}_k\}$. This is possible due

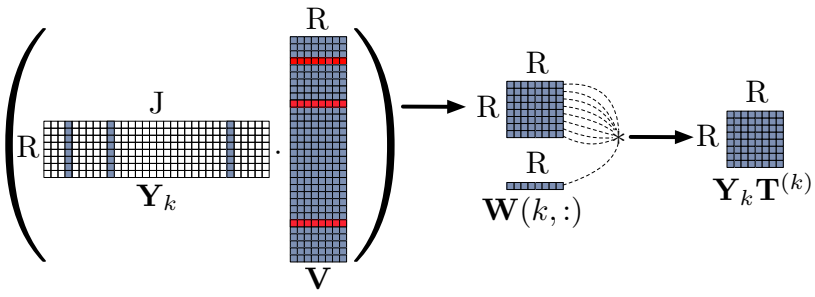



Figure 4.2: SPARTan computations for the MTTKRP w.r.t. the 1st mode. For each $k$-th partial result of Equation (4.8), we only use the rows of $\boldsymbol{V}$ factor matrix corresponding to the non-zero columns of $\boldsymbol{Y}_k$. For each of the $R$ rows of the resulting matrix, we compute the Hadamard product with $\boldsymbol{W}(k,:)$, which is the $k$-th row of the factor matrix $\boldsymbol{W}$. The described computations fulfill all of the desirable properties presented in Section 4.4.1.

to the observation that the $k$-th frontal slice $\boldsymbol{Y}_k = \boldsymbol{Q}_k^T \boldsymbol{X}_k$ of $\mathcal{Y}$ follows precisely the column sparsity pattern of $\boldsymbol{X}_k$. In particular, if $c_k$ is the number of columns of $\boldsymbol{X}_k$ containing at least one non-zero element, then $\boldsymbol{Y}_k$ will contain $R\ c_k$ non-zero elements located in the positions of the non-zero columns of $X_k$. Exploiting structured sparsity is indispensable towards minimizing intermediate data and computations to the absolutely necessary ones.

- As a by-product of the above, *SPARTan avoids unnecessary data re-organization* (tensor reshaping/permutations), since all operations are formulated w.r.t. the frontal slices $\boldsymbol{Y}_k$ of tensor $\mathcal{Y}$. In fact, our approach never forms the tensor $\mathcal{Y}$ explicitly and directly utilizes the available collection of matrices $\{\boldsymbol{Y}_k\}$ instead.

### 4.4.2 Methodology

In the following, we describe the design of our MTTKRP kernel for each one of the tensor modes. We use the notation $\boldsymbol{M}^{(i)}$ to denote the MTTKRP corresponding to the $i$-th tensor mode. Note that our factor matrices are: $\boldsymbol{H} \in \mathbb{R}^{R \times R}$, $\boldsymbol{V} \in \mathbb{R}^{J \times R}$ and $\boldsymbol{W} \in \mathbb{R}^{K \times R}$ as in Line 10 of Algorithm 5.

**Mode-1 MTTKRP.** First, we re-visit the MTTKRP equation:

$$\boldsymbol{M}^{(1)} = \boldsymbol{Y}_{(1)} \left( \boldsymbol{W} \odot \boldsymbol{V} \right), \tag{4.7}$$

where $\boldsymbol{M}^{(1)} \in \mathbb{R}^{R \times R}$, $\boldsymbol{Y}_{(1)} \in \mathbb{R}^{R \times KJ}$. In order to attempt to parallelize the above computation w.r.t. the $K$ subjects, we define the matrix $\boldsymbol{T}^{(k)} \in \mathbb{R}^{J \times R}$ to denote the $k$-th vertical block of the Khatri-Rao Product $\boldsymbol{W} \odot \boldsymbol{V} \in \mathbb{R}^{KJ \times R}$:

$$\boldsymbol{W} \odot \boldsymbol{V} = \begin{bmatrix} \boldsymbol{T}^{(1)} \\ \boldsymbol{T}^{(2)} \\ \vdots \\ \boldsymbol{T}^{(K)} \end{bmatrix}$$

We then remark that $\boldsymbol{Y}_{(1)}$ (i.e., mode-1 matricization of $\mathcal{Y}$) consists of an horizontal concatenation of the tensor's frontal slices $\boldsymbol{Y}_k$. Thus, we exploit the fact that the matrix multiplication in Equation (4.7) can be expressed as the sum of outer products or more generally, as a sum of block-by-block matrix multiplications:

$$\boldsymbol{M}^{(1)} = \sum_{k=1}^{K} \boldsymbol{Y}_k \, \boldsymbol{T}^{(k)} \tag{4.8}$$

Through Equation (4.8), the computation can be easily parallelized over $K$ independent sub-problems and then sum the partial results. This directly utilizes the frontal slices $\boldsymbol{Y}_k$ without further tensor organization. However, it constructs the whole Khatri-Rao Product (in the form of blocks $\boldsymbol{T}^{(k)}$). In order to avoid that, we first state an expression for each $i$-th row of $\boldsymbol{T}^{(k)}$, which is a direct consequence of the Khatri-Rao Product definition:

$$\boldsymbol{T}^{(k)}(i,:) = \boldsymbol{V}(i,:) * \boldsymbol{W}(k,:), \tag{4.9}$$

where $*$ stands for the Hadamard (element-wise) product. Then, we express the $j$-th row

of each partial result of Equation (4.8) as follows:

$$
\begin{aligned}
[\boldsymbol{Y}_k\, \boldsymbol{T}^{(k)}]_{j,:} &= \boldsymbol{Y}_k(j,:)\, \boldsymbol{T}^{(k)} \\
&\overset{Eq.(4.9)}{=} \sum_i \boldsymbol{Y}_k(j,i) * (\boldsymbol{V}(i,:) * \boldsymbol{W}(k,:)) \\
&\overset{(a)}{=} \left( \sum_i \boldsymbol{Y}_k(j,i) * \boldsymbol{V}(i,:) \right) * \boldsymbol{W}(k,:) \\
&\overset{(b)}{=} (\boldsymbol{Y}_k(j,:)\, \boldsymbol{V}) * \boldsymbol{W}(k,:), \qquad (4.10)
\end{aligned}
$$

where $(a)$ stems from the associative property of the Hadamard product and the fact that $\boldsymbol{W}(k,:)$ is independent of the summation and $(b)$ from the calculation of matrix multiplication as a sum of outer-products (in particular, we encounter the sub-case of vector-matrix product).

Equation (4.10) suggests an efficient way to compute the partial results of Equation (4.8), which we illustrate in Figure 4.2. First, we compute the matrix product $\boldsymbol{Y}_k\boldsymbol{V}$ and for each row of the intermediate result of size $R \times R$, we compute the Hadamard product with $\boldsymbol{W}(k,:)$. Note that, as we discussed in Section 4.3, $\boldsymbol{Y}_k$ is expected to be column-sparse in practice, thus multiplying by $\boldsymbol{V}$ uses only those rows of $\boldsymbol{V}$ corresponding to the non-zero columns of $\boldsymbol{Y}_k$. Thus, we avoid the redundant and expensive computation of the full Khatri-Rao Product. Overall, the methodology described above enjoys all of the properties described in Section 4.4.1.

**Mode-2 MTTKRP.** The methodology followed for the Mode-2 case is similar to the one described for the 1st case. We state the corresponding MTTKRP equation:

$$\boldsymbol{M}^{(2)} = \boldsymbol{Y}_{(2)}\, (\boldsymbol{W} \odot \boldsymbol{H}) \qquad (4.11)$$

where $\boldsymbol{M}^{(2)} \in \mathbb{R}^{J \times R}$, $\boldsymbol{Y}_{(2)} \in \mathbb{R}^{J \times RK}$. The main remark is that $\boldsymbol{Y}_{(2)}$ consists of an horizontal concatenation of the *transposed* frontal slices $\{\boldsymbol{Y}_k\}$ of the intermediate tensor $\mathcal{Y}$. Thus, if we denote as $\boldsymbol{T}^{(k)}$ the $k$-th vertical block of the Khatri-Rao Product $\boldsymbol{W} \odot \boldsymbol{H}$, we can

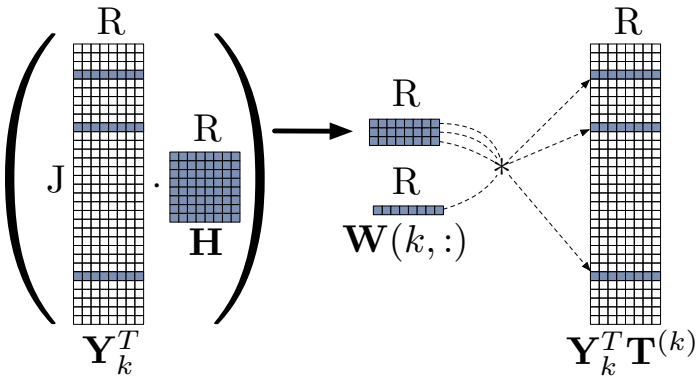



Figure 4.3: SPARTan computations for the MTTKRP w.r.t. the 2nd mode. For each $k$-th partial result of Equation (4.12), we perform the vector-matrix multiplications for each non-zero row of $\boldsymbol{Y}_k^T$. Then, for each intermediate vector, the Hadamard product with $\boldsymbol{W}(k,:)$ is computed. Finally, we distribute the vectors to their corresponding positions in $\boldsymbol{Y}_k^T\boldsymbol{T}^{(k)}$. As in the case w.r.t. the 1st mode, we limit computations to the necessary ones corresponding to the non-zero columns of $\boldsymbol{Y}_k$ and all the properties presented in Section 4.4.1 are preserved.

formulate the problem as:

$$\boldsymbol{M}^{(2)} = \sum_{k=1}^{K} \boldsymbol{Y}_k^T \, \boldsymbol{T}^{(k)} \tag{4.12}$$

Given the above, it is easy to extend Equation (4.10) for this case, so as to compute a single row of each partial result of Equation (4.12):

$$[\boldsymbol{Y}_k^T \, \boldsymbol{T}^{(k)}]_{j,:} = \left(\boldsymbol{Y}_k(:,j)^T \, \boldsymbol{H}\right) * \boldsymbol{W}(k,:) \tag{4.13}$$

The corresponding operations are illustrated in Figure 4.3. A crucial remark is that we can focus on computing the relevant intermediate results only for the non-zero rows of $\boldsymbol{Y}_{\boldsymbol{k}}^T$, since the rest of the rows of the result $\boldsymbol{Y}_{\boldsymbol{k}}^T\boldsymbol{T}^{(k)}$ will be zero. In sum, we again avoid redundant computations of the full Khatri-Rao Product and preserve all of the properties described in Section 4.4.1.

**Mode-3 MTTKRP.** First, we state the equation regarding the Mode-3 case:

$$\boldsymbol{M}^{(3)} = \boldsymbol{Y}_{(3)} \, (\boldsymbol{V} \odot \boldsymbol{H}) \tag{4.14}$$

where $\boldsymbol{M}^{(3)} \in \mathbb{R}^{K \times R}$ and $\boldsymbol{Y}_{(3)} \in \mathbb{R}^{K \times JR}$. Note that in this case, we are pursuing the MT-

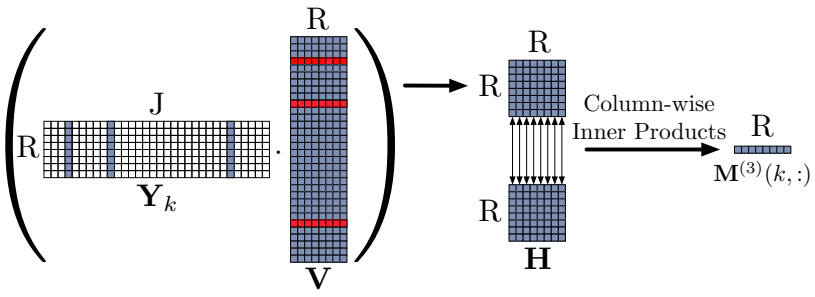



Figure 4.4: SPARTan computations for the MTTKRP w.r.t. the 3rd mode. We compute each row of the result $\boldsymbol{M}^{(3)}(k,:)$ independently of others, enabling parallelization w.r.t. the $K$ subjects. As in mode-1, mode-2 cases, we exploit the column sparsity of $\mathbf{Y}_k$. In this case, we also leverage that $\boldsymbol{H}$ is a small $R$-by-$R$ matrix in practice (due to the "size imbalance" of the intermediate tensor $\mathcal{Y}$). Thus, it is efficient to delay any computations on $\boldsymbol{H}$ until the $R$-by-$R$ product of $\mathbf{Y}_k\boldsymbol{V}$ is formed, and then take column-wise inner products between those two matrices. The described operations fulfill all the properties outlined in Section 4.4.1.

TKRP of the mode corresponding to the $K$ subjects. Thus, an entirely different approach than the Mode-1, Mode-2 cases is needed so that we construct efficient independent subproblems for each one of them. In particular, we need to design each $k$-th subproblem so that it computes the $k$-th row of $\boldsymbol{M}^{(3)}$. In addition, we want to operate only on $\{\mathbf{Y}_k\}$ without forming and reshaping the tensor $\mathcal{Y}$, as well as to exploit the frontal slices' sparsity. To tackle the challenges above, we leverage the fact that [148]:

$$\boldsymbol{M}^{(3)}(:,r) = \begin{bmatrix} \boldsymbol{H}(:,r)^T\ \mathbf{Y}_1\ \boldsymbol{V}(:,r) \\ \vdots \\ \boldsymbol{H}(:,r)^T\ \mathbf{Y}_K\ \boldsymbol{V}(:,r) \end{bmatrix} \tag{4.15}$$

Then, we remark that in order to retrieve a certain element of the matrix $\boldsymbol{M}^{(3)}$, we have:

$$\begin{aligned} \boldsymbol{M}^{(3)}(k,r) &= \boldsymbol{H}(:,r)^T\ \mathbf{Y}_k\ \boldsymbol{V}(:,r) \\ &= \boldsymbol{H}(:,r)^T\ [\mathbf{Y}_k\boldsymbol{V}](:,r) \end{aligned}$$

The last line above reflects the inner product between the corresponding $r$-th columns of $\boldsymbol{H}$ and $[\mathbf{Y}_k\ \boldsymbol{V}]$, respectively. Thus, in order to retrieve a row $\boldsymbol{M}(k,:)$, we can simply operate

as:

$$\boldsymbol{M}^{(3)}(k,:) = dot\left(\boldsymbol{H}, \boldsymbol{Y}_k\boldsymbol{V}\right) \tag{4.16}$$

where the $dot()$ function extracts the inner product of the corresponding columns of its two matrix arguments. We illustrate this operation in Figure 4.4. Since $\boldsymbol{H}$ is a small $R$-by-$R$ matrix (due to the tensor's "size imbalance"), it is very efficient to delay any computations on $\boldsymbol{H}$ until the $R$-by-$R$ intermediate matrix is formed as a product of $\boldsymbol{Y}_k$ $\boldsymbol{V}$. Then, we simply take the column-wise inner products between those two $R$-by-$R$ matrices. In that way, all the desirable properties we mentioned in Section 4.4.1 are also fulfilled.

In Algorithm 6, we list the pseudocode corresponding to the methodology proposed. Note that in lines 8,16, we can accumulate over the partial results in parallel, since the summation is independent of the iteration order.

**Algorithm 6** MTTKRP for SPARTan

**Input:** $\{\boldsymbol{Y}_k \in \mathbb{R}^{R\times J}\}$ for $k = 1, \ldots, K$, $\boldsymbol{H} \in \mathbb{R}^{R\times R}$, $\boldsymbol{V} \in \mathbb{R}^{J\times R}$, $\boldsymbol{W} \in \mathbb{R}^{K\times R}$, the target rank $R$ and the mode $n$ for which we are computing the MTTKRP
**Output:** $\boldsymbol{M}^{(n)}$

1: Initialize $\boldsymbol{M}^{(n)}$ with zeros
2: **if** $n == 1$ **then**
3:  **for** $k = 1, \ldots, K$ **do**
4:  $temp \leftarrow \boldsymbol{Y}_k V$
5:  **for** $r = 1, \ldots, R$ **do**
6:  $temp(r,:) \leftarrow temp(r,:) * \boldsymbol{W}(k,:)$
7:  **end for**
8:  $\boldsymbol{M}^{(1)} \leftarrow \boldsymbol{M}^{(1)} + temp$  *// sum in parallel* $\forall k = 1, \ldots, K$
9:  **end for**
10: **else if** $n == 2$ **then**
11:  **for** $k = 1, \ldots, K$ **do**
12:  Initialize $temp \in \mathbb{R}^{J\times R}$ with zeros
13:  **for** each $j$-th non-zero column of $\boldsymbol{Y}_k$ **do**
14:  $temp(j,:) \leftarrow \left(\boldsymbol{Y}_k(:,j)^T \boldsymbol{H}\right) * \boldsymbol{W}(k,:)$
15:  **end for**
16:  $\boldsymbol{M}^{(2)} \leftarrow \boldsymbol{M}^{(2)} + temp$  *// sum in parallel* $\forall k = 1, \ldots, K$
17:  **end for**
18: **else if** $n == 3$ **then**
19:  **for** $k = 1, \ldots, K$ **do**
20:  $\boldsymbol{M}^{(3)}(k,:) \leftarrow dot\left(\boldsymbol{H}, \boldsymbol{Y}_k\boldsymbol{V}\right)$  *// in parallel* $\forall k = 1, \ldots, K$
21:  **end for**
22: **end if**

Table 4.3: Summary statistics for the real datasets of our experiments. $K$ is the number of subjects, $J$ is the number of variables, $I_k$ is the number of observations for the $k$-th subject and #nnz corresponds to the total number of non-zeros.

| Dataset | $K$ | $J$ | $max(I_k)$ | #nnz |
|---|---|---|---|---|
| CHOA | 464,900 | 1,328 | 166 | 12.3 Mil. |
| MovieLens | 25,249 | 26,096 | 19 | 8.9 Mil. |

## 4.5 Experiments

### 4.5.1 Setup

**Real Data Description.** Table 4.3 provides summary statistics regarding the real datasets used.

The **CHOA** (Children Healthcare of Atlanta) dataset corresponds to EHRs of pediatric patients with at least 2 hospital visits. For each patient, we utilize the diagnostic codes and medication categories from their records, as well as the provided age of the patient (in days) at the visit time. The available International Classification of Diseases (ICD9) [43] codes are summarized to Clinical Classification Software (CCS) [111] categories, which is a standard step in healthcare analysis improving interpretability and clinical meaningfulness. We aggregate the time mode by week and all the medical events over each week are considered as a single observation. The resulting data are of 464,900 subjects by 1,328 features by maximum 166 observations with 12.3m non-zeros.

**MovieLens** 20M is another real dataset we used, which is *publicly available* [2]. We are motivated to use this dataset, because of the importance of the evolution of user preferences over time, as highlighted in recent literature [156]. For this dataset, we consider that each year of ratings corresponds to a certain observation; thus, for each user, we have a year-by-movie matrix to describe her rating activity. We consider only the users having at least 2 years of ratings.

**Implementation details.** We used MatlabR2015b for our implementations, along with

[2] `https://grouplens.org/datasets/movielens/`

functionalities for sparse tensors from the Tensor Toolbox [114] and the Non-Negative Least Squares (NNLS) approach [146] from the N-way Toolbox [157] [3]. In both the SPARTan and the baseline implementations, we adjust the CP-ALS iteration arising in the PARAFAC2-ALS, so that non-negative constraints are imposed on the $\{\boldsymbol{S}_k\}, \boldsymbol{V}$ factors, as discussed in Section 4.3.2.

**The baseline method** corresponds to the standard fitting algorithm for the PARAFAC2 model [59] adjusted for sparse tensors as in [144]. We utilized the implementation from the most recent version of the Tensor Toolbox [114] regarding both the manipulation of sparse tensors, as well as the CP-ALS iteration arising in the PARAFAC2-ALS.

**Parallelism.** We exploit the capabilities of the Parallel Computing Toolbox of Matlab, by utilizing its parallel pool in both SPARTan and the baseline approach, whenever this is appropriate. Regarding the size of the parallel pool, the number of workers of all the experiments regarding a certain dataset is fixed. For the movie-rating dataset we used the default of $12$ workers. For the synthetic and the CHOA datasets, we increased the number of workers to $20$ because of the data size increase.

**Hardware.** We conducted our experiments on a server running Ubuntu 14.04 with 1TB of RAM and four Intel E5-4620 v4 CPU's with a maximum clock frequency of 2.10GHz. Each one of the processors contains $10$ cores, and each one of the cores can exploit $2$ threads with hyper-threading enabled.

### 4.5.2 SPARTan is fast and memory-efficient

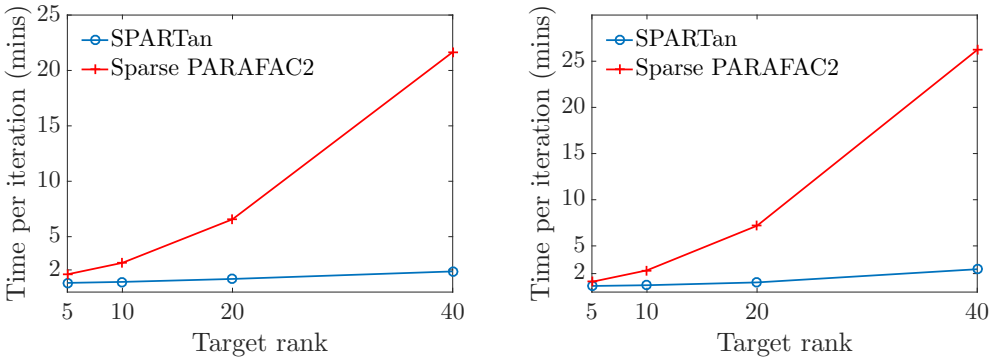
**Synthetic Data.** We assess the scalability of the approaches under comparison for sparse synthetic data. We considered a setup with $1,000,000$ subjects, $5,000$ variables and a maximum of $100$ observations for each subject. The number of observations $I_k$ for each subject is dependent on the number of rows of $\boldsymbol{X}_k$ containing non-zero elements; thus, $I_k$ increases with the dataset density. Indicatively, the mean number of observations $I_k$ for the

[3] We also accredit the dense PARAFAC2 implementation by Rasmus Bro, from where we have adapted many functionalities.

sparsest dataset created ($\approx$ 63 mil.) is $46.9$ and for the densest ($\approx$ 500 mil.) dataset, the mean $I_k$ is $99.3$. We randomly construct the factors of a rank-$40$ (which is the maximum target rank used in our experiments) PARAFAC2 model. Based on this model, we construct the input slices $\{\boldsymbol{X}_k\}$, which we then sparsify uniformly at random, for each sparsity level. The density of the sparsification governs the number of non-zeros of the collection of input matrices.

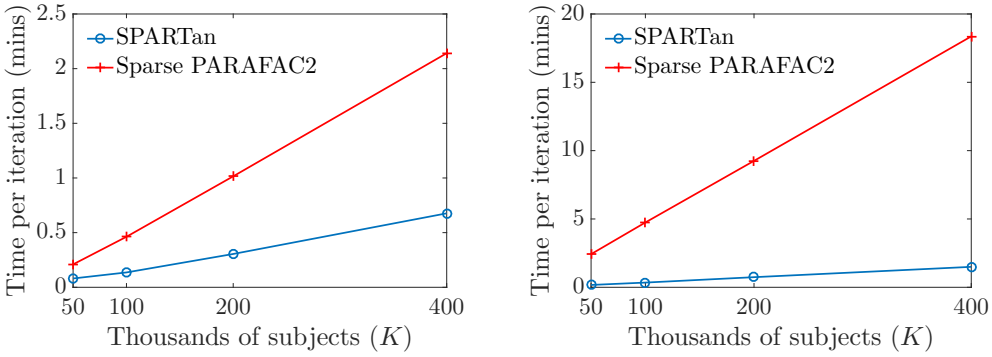
We provide the results in Table 4.1. First, we remark that SPARTan is *both more scalable* and *faster* than the baseline. In particular, the baseline approach fails to execute in the two largest problem instances for target rank $R = 40$, due to out of memory problems, during the creation of the intermediate sparse tensor $\mathcal{Y}$. Note that as we discussed in Section 4.4, SPARTan avoids the additional overhead of explicitly constructing a sparse tensor structure, since it only operates directly on the tensor's frontal slices $\{\boldsymbol{Y}_k\}$. Regarding the baseline's memory issue, since the density of $\mathcal{Y}$ may grow (e.g., $\approx 10\%$ in the densest case), we also attempted to store the intermediate tensor $\mathcal{Y}$ as a dense one. However, this also failed, since the memory requested for a dense tensor of size $40$-by-$5$K-by-$1$Mil. exceeded the available RAM of our system (1TB). Overall, it is clear that the baseline approach cannot fully exploit the input sparsity. On the contrary, SPARTan properly executes for all the problem instances considered in a reasonable amount of time. In particular, for $R = 40$, SPARTan is up to $22\times$ faster than the baseline. Even for a lower target rank of $R = 10$, SPARTan achieves up to $13\times$ faster computation.

**Real Data.** We evaluate the scalability of the proposed SPARTan approach against the baseline method for the real datasets as well. In Figures 4.5, 4.6, 4.7, we present the results of the corresponding experiments. First, we target the full datasets and vary the pursued target rank (Figure 4.5). Note that for both datasets considered, the time per iteration of the baseline approach increases dramatically as we increase the target rank. On the contrary, the time required by SPARTan increases only slightly. Overall, our approach achieves up to over an order of magnitude gain regarding the time required per epoch for both datasets.



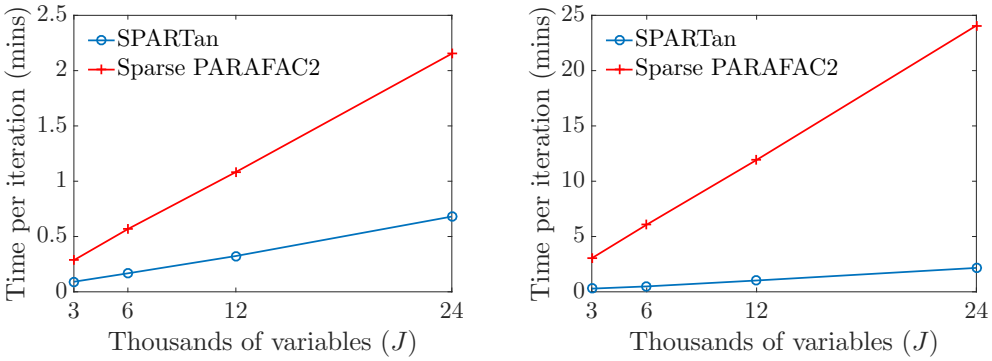

(a) CHOA dataset (b) MovieLens dataset

Figure 4.5: Time in minutes for one iteration (as an average over 10) for varying target rank for both the real datasets used. SPARTan achieves up to $12\times$ and $11\times$ speedup over the baseline approach for the CHOA and the MovieLens datasets respectively.




(a) Target Rank $R = 10$ (b) Target Rank $R = 40$

Figure 4.6: CHOA dataset: Time in minutes for one iteration (as an average over 10) for varying number of subjects ($K$) included and fixed target rank (two cases considered: $R = \{10, 40\}$).




(a) Target Rank $R = 10$ (b) Target Rank $R = 40$

Figure 4.7: MovieLens dataset: Time in minutes for one iteration (as an average over 10) for varying number of variables ($J$) included and fixed target rank (two cases considered: $R = \{10, 40\}$).

We also evaluate the scalability of the methods under comparison as we vary the subjects and the variables considered. Since the CHOA dataset (Figure 4.6) contains more subjects than variables, we vary the number of subjects for this dataset for two fixed target ranks $(10, 40)$. In both cases, SPARTan scales better than the baseline. As concerns the MovieLens dataset (Figure 4.7), since it contains more variables than subjects, we examine the scalability w.r.t. increasing subsets of variables considered. In this case as well, we remark the favorable scalability properties of SPARTan, rendering it practical to use for large and sparse "irregular" tensors.

### 4.5.3 Phenotype discovery on CHOA EHR Data

**Motivation.** Next we demonstrate the usefulness of PARAFAC2 towards temporal phenotyping of EHRs. Phenotyping refers to the process of extracting meaningful patient clusters (i.e., phenotypes) out of raw, noisy Electronic Health Records [45]. An open challenge in phenotyping is to capture temporal trends or patterns regarding the evolution of those phenotypes for each patient over time. Below, we illustrate how SPARTan can be used to successfully tackle this challenge.

**Model Interpretation:** We propose the following model interpretation towards the target challenge:

- The common factor matrix $\boldsymbol{V}$ reflects the *phenotypes' definition* and the non-zero values of each $r$-th column indicate the membership of the corresponding medical feature to the $r$-th phenotype.
- The diagonal $\boldsymbol{S}_k$ provides the *importance membership indicators* of the $k$-th subject to each one of the $R$ phenotypes/clusters. Thus, we can sort the $R$ phenotypes based on the values of vector $diag(\boldsymbol{S}_k)$ and identify the most relevant phenotypes for the $k$-th subject.
- Each $\boldsymbol{U}_k$ factor matrix provides the *temporal signature* of each patient: each $r$-th column

of $\boldsymbol{U}_k$ reflects the evolution of the expression of the $r$-th phenotype for all the $I_k$ weeks of her medical history. Note that since all $\boldsymbol{X}_k, \boldsymbol{S}_k, \boldsymbol{V}$ matrices are non-negative, we only consider the non-negative elements of the temporal signatures in our interpretation.

**Temporal Phenotyping of Medically Complex Patients (MCPs)** In order to illustrate the use of PARAFAC2 towards temporal phenotyping, we focus our analysis on a subset of pediatric patients from CHOA, which are classified by them as Medically Complex. These are the patients with high utilization, multiple specialty visits and high severity. Conceptually, those patients suffer from chronic and/or very severe conditions that are hard to treat. As a result, it becomes a very important challenge to accurately phenotype those patients, as well as provide a *temporal signature* for each one of them, which summarizes their phenotypes' evolution.

The number of MCPs in the CHOA cohort is $8,044$, their diagnoses and medications sum up to $1,126$, and the mean number of weekly observations for those patients is $28$. We ran SPARTan for target rank $R = 5$ and the phenotypes discovered are provided in Table 4.4 (*phenotypes' definition* matrix). The labels for each group are the definitions of the phenotypes provided by the medical expert, who endorsed their clinical meaningfulness.

In Figure 4.8, we provide part of the real EHR, as well as the temporal signature produced by SPARTan, for a certain medically complex patient. Regarding the EHR, we visualize the subset of diagnoses and medications for which the sum of occurrences for the whole patient history is above a certain threshold (e.g., 5 occurrences). This step ensures that the visualized EHR will only contain the conditions exhibiting some form of temporal evolution. For the patient example considered, we identify the top-2 relevant phenotypes through the importance membership indicator matrix $\boldsymbol{S}_k$ as discussed above. For those top-2 phenotypes, we present the resulting *temporal signature*, from which we easily detect intricate temporal trends of the phenotypes involved. Those trends were confirmed by the clinical expert as valuable towards fully understanding the phenotypic behavior of the MCPs.

Table 4.4: Phenotypes discovered by PARAFAC2. The title annotation for each phenotype is provided by the medical expert. The red color corresponds to diagnoses and the blue color corresponds to medications.

| **Cancer** | **Weight** |
|---|---|
| Chemotherapy | 0.35 |
| Leukemias [39.] | 0.27 |
| Immunity disorders [57.] | 0.23 |
| HEPARIN AND RELATED PREPARATIONS | 0.6 |
| ANTIEMETIC/ANTIVERTIGO AGENTS | 0.34 |
| SODIUM/SALINE PREPARATIONS | 0.32 |
| TOPICAL LOCAL ANESTHETICS | 0.19 |
| ANTIHISTAMINES - 1ST GENERATION | 0.16 |
| **Sickle Cell Anemia (SCA)** | **Weight** |
| Sickle cell anemia [61.] | 0.73 |
| NSAIDS, CYCLOOXYGENASE INHIBITOR - TYPE | 0.31 |
| ANALGESICS NARCOTICS | 0.26 |
| FOLIC ACID PREPARATIONS | 0.2 |
| BETA-ADRENERGIC AGENTS | 0.18 |
| SODIUM/SALINE PREPARATIONS | 0.16 |
| **Neurological System Disorders** | **Weight** |
| Other nervous system symptoms and disorders | 0.56 |
| Rehabilitation care; fitting of prostheses; and adjustment of devices [254.] | 0.5 |
| Residual codes; unclassified; all E codes [259. and 260.] | 0.46 |
| Other connective tissue disease [211.] | 0.33 |
| Other and unspecified metabolic; nutritional; and endocrine disorders | 0.18 |
| **Gastrointestinal Disorders** | **Weight** |
| Residual codes; unclassified; all E codes [259. and 260.] | 0.2 |
| Other and unspecified metabolic; nutritional; and endocrine disorders | 0.15 |
| Other and unspecified gastrointestinal disorders | 0.15 |
| ANALGESIC/ANTIPYRETICS NON-SALICYLATE | 0.32 |
| POTASSIUM REPLACEMENT | 0.26 |
| BETA-ADRENERGIC AGENTS | 0.23 |
| ANALGESICS NARCOTICS | 0.23 |
| SODIUM/SALINE PREPARATIONS | 0.22 |
| SEDATIVE-HYPNOTICS NON-BARBITURATE | 0.21 |
| ANTIEMETIC/ANTIVERTIGO AGENTS | 0.19 |
| ANALGESICS NARCOTIC ANESTHETIC ADJUNCT AGENTS | 0.18 |
| NSAIDS, CYCLOOXYGENASE INHIBITOR - TYPE | 0.16 |
| IRRIGANTS | 0.16 |
| LAXATIVES AND CATHARTICS | 0.15 |
| GENERAL INHALATION AGENTS | 0.15 |
| **Liver/Kidney System Disorders** | **Weight** |
| Other aftercare [257.] | 0.8 |
| hronic kidney disease [158.] | 0.39 |
| Other and unspecified liver disorders | 0.3 |
| Immunity disorders [57.] | 0.16 |

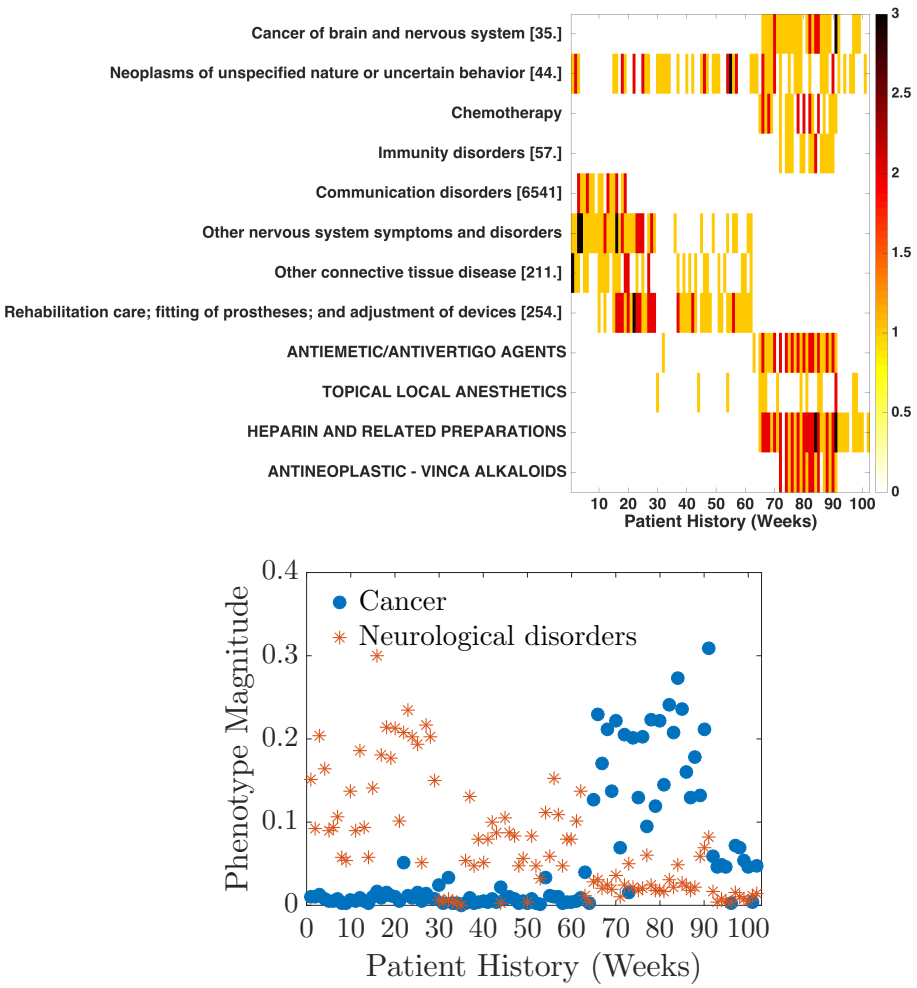



Figure 4.8: *Upper part:* Part of real EHR data of a Medically Complex Patient (MCP). For each week, it contains the occurrences of a diagnosis/medication in the patient's records. *Lower part:* Temporal signature of the patient created by SPARTan. PARAFAC2 captures the stage where cancer treatment is initiated (week 65). At that point, indications of cancer treatment and diagnosis, such as cancer of brain, chemotherapy, heparin and antineoplastic drugs start to get recorded in the patient history. PARAFAC2 also captures the presence of neurological disorders during the first weeks of the patient history. The definition for each phenotype as produced by PARAFAC2 can be found in Table 4.4.

## 4.6 Discussion & Conclusions

PARAFAC2 has been the state-of-the-art model for mining "irregular" tensors, where the observations along one of its modes do not align naturally. However, it has been highly disregarded by practitioners, as compared to other tensor approaches. Bro [143] has summarized the reason for that as:

> *The PARAFAC2 model has not yet been used very extensively maybe because the implementations so far have been complicated and slow.*

The methodology proposed in this work renders this statement no longer true for large and sparse data. In particular, as tested over real and synthetic datasets, SPARTan is both fast and memory-efficient, achieving up to $22\times$ performance gains over the best previous implementation and also handling larger problem instances for which the baseline fails due to insufficient memory.

The key insight driving SPARTan's scalability is the pursuit and exploitation of special structure in the data involved in intermediate computations; prior art did not do so, instead treating those computations as a black-box.

The capability to run PARAFAC2 at larger scales is, in our view, an important enabling technology. As shown in our evaluations on EHR data, the clinically meaningful phenotypes and temporal trends identified by PARAFAC2 reflect the ease of the model's interpretation and its potential utility in other application domains.

Future directions include, but are not limited to: *a)* development of PARAFAC2 algorithms for alternative models of computation, such as distributed clusters [147], or supercomputing environments; *b)* extension of the methodology proposed for higher-order "irregular" tensors with more than one mismatched mode.

Finally, to enable reproducibility and promote further popularization of the PARAFAC2 modeling within the area of data mining, we open-source our implementations and make them *publicly available*.

# CHAPTER 5

# TEMPORAL PHENOTYPING OF MEDICALLY COMPLEX CHILDREN VIA PARAFAC2 TENSOR FACTORIZATION


Summary

How tensor factorization can be applied and extended to distill the EHR data of a cohort of patients facing complex clinical manifestations into clinically-meaningful phenotype definitions, their temporal trends over time and associated patient groups? In this Chapter, we demonstrate how the PARAFAC2 model, as implemented in Chapter 4, can be applied and extended to extract phenotype definitions of medically-complex children from Children's Healthcare of Atlanta. We identified 4 phenotypes which are validated by a clinical expert, as well as significant survival variations among different phenotypes.


**Objective:** Our aim is to extract clinically-meaningful phenotypes from longitudinal electronic health records (EHRs) of medically-complex children. This is a fragile set of patients consuming a disproportionate amount of pediatric care resources but who often end up with sub-optimal clinical outcome. The rise in available electronic health records (EHRs) provide a rich data source that can be used to disentangle their complex clinical conditions into concise, clinically-meaningful groups of characteristics. We aim at identifying those phenotypes and their temporal evolution in a scalable, computational manner, which avoids the time-consuming manual chart review.

**Materials and Methods:** We analyze longitudinal EHRs from Children's Healthcare of Atlanta including 1,045 medically complex patients with a total of 59,948 encounters over 2 years. We apply a tensor factorization method called PARAFAC2 to extract: a) clinically-meaningful groups of features b) concise patient representations indicating the presence of

a phenotype for each patient, and c) temporal signatures indicating the evolution of those phenotypes over time for each patient.

**Results:** We identified four medically complex phenotypes, namely gastrointestinal disorders, oncological conditions, blood-related disorders, and neurological system disorders, which have distinct clinical characterizations among patients. We demonstrate the utility of patient representations produced by PARAFAC2, towards identifying groups of patients with significant survival variations. Finally, we showcase representative examples of the temporal phenotypic trends extracted for different patients.

**Discussion:** Unsupervised temporal phenotyping is an important task since it minimizes the burden on behalf of clinical experts, by relegating their involvement in the output phenotypes validation. PARAFAC2 enjoys several compelling properties towards temporal computational phenotyping: a) it is able to handle high-dimensional data and variable numbers of encounters across patients, b) it has an intuitive interpretation and c) it is free from ad-hoc parameter choices. Computational phenotypes, such as the ones computed by our approach, have multiple applications; we highlight three of them which are particularly useful for medically complex children: 1) integration into clinical decision support systems, 2) interpretable mortality prediction and 3) clinical trial recruitment.

**Conclusion:** PARAFAC2 can be applied to unsupervised temporal phenotyping tasks where precise definitions of different phenotypes are absent, and lengths of patient records are varying.

## 5.1 Objective

Medically complex or fragile children need intensive medical care due to multisystem dysfunction, technology dependence, or complex medication needs.[158] It has been estimated that children with special healthcare needs constitute about 18% (near 12.6 millions) of all the US children. [159] Their population is also growing at approximately 5% annually, outpacing the rate of growth of typically developing children (1-2%).[160] Given the large

portion of complex children in consumption of hospital resources,[161] the Institute of Medicine has identified Children with Special Health Care Needs (CSHCN) as a priority population for study.[162] Medically complex patients consume significant disproportionate amount of care resources in hospitals.[161] For example, the Children's Hospital Association reported that among the 33 million children in Medicaid in 2013, the 2 million with medical complexity were in need of approximately 10 times the costs per year, on average, compared to other children.[163]

Important ongoing research is on how to improve the outcome such as survival and decrease the healthcare costs of medically complex children. In order to achieve such advancements, there have been several efforts reported in predicting hospital readmission [164] as well as exposing the increase in medically complex children population. [158] With the advent and availability of EHRs, it is possible to extract computational phenotypes to group medically complex children based on their underlying heterogeneity. Each one of the groups is described by a clinically-meaningful set of features, i.e., the phenotype definitions.

Phenotyping based on manual chart review is time-consuming and not scalable to large patient cohorts. [38, 37] There have been several efforts to extract computational phenotypes based on rule-based algorithms, separating patients based on carefully crafted feature sets and their possible ranges. The eMERGE network [165] is such a leading effort and the PheKB repository contains algorithms from eMERGE and other sources. [166] The main drawback of such approaches is the substantial burden and effort on both medical informatics experts and clinicians to develop specialized criteria for each individual phenotype. As such, these approaches are notoriously hard to scale for increasing number of patients and heterogeneous patient cohorts. [167]

To tackle the above challenges, it is imperative to establish accurate and efficient methods for computational phenotyping through readily available EHR data. Computational phenotypes can then be used in multiple applications [7], such as in clinical decision sup-

port [37], clinical predictive modeling [24] and automatically determining eligibility for clinical trial recruitment [5]. In this work, in the absence of expert-annotated labels for medically-complex children, we focus on unsupervised computational phenotyping. We are additionally interested in extracting information revealing when a certain phenotype occurred; such insights may lead to a better understanding of complex disease progression (a task we call as temporal phenotyping [10]). To do so, we have to overcome the following challenges, pertaining to the nature of the EHRs:

- High-dimensionality and incompleteness of EHR records. EHRs are typically high-dimensional event sequences, including information from different modalities, such as diagnoses, medications and procedures. Moreover, the recording of EHRs is very far from the rigorous data collection employed in formal experiments, [168] since records only occur during patient encounters at the hospital; thus, sparsity and discontinuities of clinical features are a commonplace.[169]

- Variable number of medical encounters across patients. Another important characteristic of EHRs is the variable number of observations across patients. This renders any temporal alignment for analysis purposes an intricate task. In fact, even if two patients share the same length of records, their temporal alignment is not necessarily meaningful for many reasons, such as: a) disease stages and progression vary, and b) more records are present when a patient is undergoing a more severe condition, thus we should not consider the observations as random samples over time.[170]

To successfully tackle these challenges, we apply and extend the PARAFAC2 tensor factorization in order to extract phenotype descriptions from medically complex children, associated patient groups and temporal trends of phenotypes. PARAFAC2 [58] can handle high-dimensional, sparse and irregular input data. It can also accept a variable number of observations for each patient and provides a great match for using sequence time as the time parameterization of EHR data. [53, 170] To frame the input for PARAFAC2, we use a

binary occurrence matrix of size $I_k \times J$ reflecting the $k$-th patient's medical history, where $I_k$ corresponds to the $k$-th patient's number of encounters and $J$ to the number of medical features. Then, we apply PARAFAC2 [58] to the collection of matrices $\{\boldsymbol{X}_k \in \mathbb{R}^{I_k \times J}\}$ which leads to the following approximation:

$$\boldsymbol{X}_k \approx \boldsymbol{U}_k \, \boldsymbol{S}_k \, \boldsymbol{V}^T$$

where $k = 1, \ldots, K, \boldsymbol{U}_k \in \mathbb{R}^{I_k \times R}, \boldsymbol{S}_k \in \mathbb{R}^{R \times R}$ is diagonal and $\boldsymbol{V} \in \mathbb{R}^{J \times R}$. Through this model, we extract: a) the definition of the R phenotypes ($\boldsymbol{V}$ matrix in the model), b) a personalized phenotype membership indicator for each patient reflecting the existence/absence of each phenotype within her medical history ($\boldsymbol{S}_k$ matrix in the model), c) a personalized profile for each patient reflecting the phenotypic evolution across her encounters ($\boldsymbol{U}_k$ matrix in the model). In contrast to other methodologies proposed to tackle temporal phenotyping, PARAFAC2 preserves uniqueness of the output model. [59] In non-unique models, there may be many arbitrarily rotated versions of a solution yielding the same approximation error and it is unclear to the practitioner which one of those solutions provides the true latent factors. Unique models avoid such rotational ambiguities, thus boosting interpretability.

In addition to recent work proposing advancements on the scalability of PARAFAC2 for EHR data [51], in this work we illustrate an automatic way of determining the number of phenotypes. To do so, we apply the Core Consistency Diagnostic (CORCONDIA) to automatically discover the number of phenotypes required to succinctly summarize the different phenotypes of medically complex patients. [171] This step makes PARAFAC2 free from ad-hoc parameter choices. We also present how patients can be grouped based on the learned model parameters.

With PARAFAC2, we identified four medically complex phenotypes, namely gastrointestinal disorders, oncological conditions, blood-related disorders, and neurological system disorders, which have distinct clinical characterizations among patients. We demonstrate

the utility of patient representations produced by PARAFAC2, towards identifying groups of patients with significant survival variations. Finally, we showcase representative examples of the temporal phenotypic trends extracted for different patients.

## 5.2 Background and Significance

Below, we review previous approaches related to unsupervised temporal phenotyping. For a broader discussion on phenotyping based on EHRs, including rule-based and supervised phenotyping approaches, we refer to existing survey papers. [7, 172]

**Probabilistic Approaches**

Lasko et al. study the temporal phenotyping problem from a probabilistic perspective (using Gaussian Processes).[173] In contrast to our application, they focus on continuous variables (e.g. uric acid concentration over time). The sparsity and irregularities of medical codes make their approach difficult to handle high-dimensional discrete event sequences (e.g., diagnoses and medications). Wang et al. use a Markov Jump Process to model the stage transitions of a certain disease, as reflected by a set of comorbidities.[8] Their methodology targets a single disease and requires a homogeneous cohort as input (sharing the target disease). Also, their modeling assumes a certain set of hidden disease stages which is shared among all the comorbidities; this assumption may be restrictive for medically complex patients who often share multiple different evolving phenotypes. Similar specialized modeling of a single conditions progression has been studied in [174], where the authors focus on extracting temporal trajectories of individual input variables, rather than the trajectories of clinically-meaningful feature groups. Another probabilistic approach is proposed by Ghassemi et al.,[175] where Multi-Task Gaussian Processes are used to model the temporal evolution of clinical concepts. The main limitations of that approach are the high computational cost and the multi-modal hyper-parameter space to be tuned; in contrast, our approach does not require any ad-hoc parameter search. Static probabilistic phenotyping approaches have also been proposed (e.g., the UPhenome model [176]). However,

those do not encode any temporality about the patient records.

**Non-unique Factorization-Based Temporal Phenotyping**

The idea of factorizing temporally-evolving matrices of clinical features has already appeared in the literature. Wang et al. propose a convolutional framework of matrix factorization to capture temporal phenotypic trends.[138] In other prior art, matrices of clinical features over time are jointly factorized to capture phenotypes and their evolution.[139, 177, 178] In contrast to those works, our PARAFAC2 application provides model uniqueness; thus, reliable interpretation of the resulting phenotypes and phenotypic trends is possible, avoiding rotational ambiguities of the output model.

**Static Matrix/Tensor Phenotyping**

Static computational phenotyping involving tensors [134, 135, 179, 180, 48] and matrices [46, 181] has also been an active subject of interest. However, those works often avoid modeling time directly via temporal aggregation. However, as we show in this work, the temporal evolution of phenotypes can provide crucial insights regarding disease progression.

**Clustering Approaches**

There do exist other unsupervised approaches, such as k-means or hierarchical clustering, which can be used to group patients based on a set of clinical features. [182] However, such approaches only provide the patient clusters, without the corresponding definitions of the phenotypes that could explain the reasoning behind the groupings, as happens in PARAFAC2. At the same time, PARAFAC2 provides a temporal signature indicating the evolution of phenotypes for each patient. Simpler clustering approaches preserve none of the above properties. They may additionally require preprocessing of the input longitudinal data into segments, so that those are comparable across patients. [183, 184] Our approach based on PARAFAC2 enables avoiding such potentially-limiting assumptions, which are usually disease-dependent and may obfuscate the extracted temporal patterns.

**Embedding-based approaches**

The idea of word embeddings [185] has been used to extract vector representations of medical concepts (e.g., diagnostic codes) and patients. The success of word embeddings is attributed to the fact that contextually similar terms have their corresponding vectors close to each other in the embedding space. There do exist extensions of word embeddings extracting medical concept and visit, [186] as well as patient vector representations. [167] However, these works do not focus on extracting the phenotype trajectories over time.

**Careflow mining**

Approaches based on sequential pattern mining have also been proposed to model the temporality of care provided. [10, 187] However, those approaches extract sequences of individual events which capture a certain flow of care, rather than evolving phenotypic definitions based on shared groups of features among different visits and patients. Also, pattern mining methods do not provide computable low-dimensional patient representations based on the extracted phenotypes; these can be further used for clustering and prediction tasks.

**Association analysis**

Finally, there do exist works which conduct association analysis of diagnostic codes and visualize the resulting association patterns. [188] However, these methods only provide a high-level understanding of diagnostic code associations over time, without neither providing precise phenotypic definitions nor identifying the importance of different phenotypes for every patient.

## 5.3 Materials and Methods

### Input Data Formulation

First, we formally describe the input data representation. A natural way to represent the EHR of each k-th patient is to use a binary matrix $X_k$ of size $I_K \times J$ to model the $I_k$ encounters from patient k about J clinical features. If we observe the j-th clinical feature (e.g., diagnosis of respiratory disorders) recorded during the i-th encounter of the patient ks records, then this is represented as $X_k(i, j) = 1$. In Figure 5.1, we provide a visual-

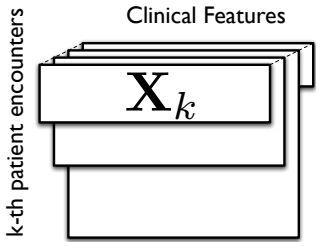



Figure 5.1: Input data representation before temporal phenotyping: each slice $X_k$ represents the data from k-th patient, each row corresponds to one encounter from that patient, each column represents a particular clinical feature (e.g., whether diagnosis of brain cancer is present or not). All patients shared the same set of clinical features (i.e., the same number of columns) but with potentially different number of encounters (e.g., variable numbers of rows).

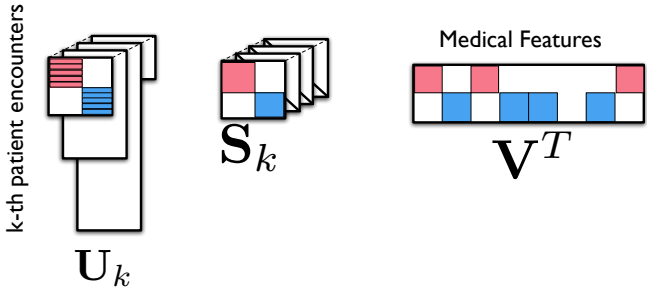



Figure 5.2: PARAFAC2 Decomposition example for temporal phenotyping with R=2 phenotypes pursued

ization of the input data. Note that we are modeling three input modes: patients, clinical features, patient encounters. Given such unique input representation, the most popular tensor methods such as Canonical Polyadic (CP) [23] and Tucker [73] have to assume the same number of encounters across all patients, which is not true in practice. Note that an attempt to collapse the patient mode by vertically concatenating all the encounters into a single matrix would result in abandoning the patient-level information; as we will demonstrate in the experiments, this information is crucial in identifying patient sub-groups with similar phenotypic characteristics. We next describe how PARAFAC2 [58] can tackle this challenge.

Figure 5.2 provides an overview of the PARAFAC2 decomposition [58] towards temporal phenotyping. PARAFAC2 approximates a collection of matrices $\{\boldsymbol{X}_k \in \mathbb{R}^{I_k \times J}\}$ as:

$$\boldsymbol{X}_k \approx \boldsymbol{U}_k \, \boldsymbol{S}_k \, \boldsymbol{V}^T \tag{5.1}$$

where $k = 1, \ldots, K, \boldsymbol{U}_k \in \mathbb{R}^{I_k \times R}$, $\boldsymbol{S}_k \in \mathbb{R}^{R \times R}$ is diagonal and $\boldsymbol{V} \in \mathbb{R}^{J \times R}$.

Intuitively, the output components from PARAFAC2 model provide the key towards temporal phenotyping: a) the definition of the $R$ phenotypes ($\boldsymbol{V}$ matrix in the model), b) a personalized phenotype membership indicator for each patient reflecting the existence/absence of each phenotype within her medical history ($\boldsymbol{S}_k$ matrix in the model), c) a personalized profile for each patient reflecting the phenotypic evolution across her encounters ($\boldsymbol{U}_k$ matrix in the model).

The model can be extended so that non-negative constraints are imposed on $\{\boldsymbol{S}_k\}, \boldsymbol{V}$ factors.[189] To further enhance interpretability, we employ this constraint in our experiments and given that the input data are non-negative (binary), we elaborate our interpretation in more details:

- The common factor matrix $\boldsymbol{V}$ reflects the *phenotypes' definition* and the non-zero values of each $r$-th column indicate the membership of the corresponding medical feature to the $r$-th phenotype.

- The diagonal $\boldsymbol{S}_k$ provides the *importance membership indicators* of the $k$-th subject to each one of the $R$ phenotypes/clusters. Thus, we can sort the $R$ phenotypes based on the values of vector $diag(\boldsymbol{S}_k)$ and identify the most relevant phenotypes for the $k$-th subject.

- Each $\boldsymbol{U}_k$ factor matrix provides the *temporal signature* of each patient: each $r$-th column of $\boldsymbol{U}_k$ reflects the evolution of the expression of the $r$-th phenotype for all the $I_k$ daily encounters of her medical history.

**Model and algorithm details**

PARAFAC2 preserves uniqueness of the model output, by imposing the constraint that the

cross product $\mathbf{U}_k^T\mathbf{U}_k$ is invariant regardless which k is involved. [58] This implies that the correlations between the phenotypes are kept constant, for all the encounters. [60] Preserving uniqueness means that the model is pursuing the actual latent factors, rather than an arbitrary rotation of them.[131] The issue with non-unique models is that many arbitrarily rotated decompositions may yield equivalent solutions in terms of the approximation error. As a result, it is unclear to the practitioner which one of those solutions does provide the true latent factors. Overall, in applications where we care about model interpretation (as happens in temporal phenotyping) and not merely about compressing the input data, unique models have a clear advantage over non-unique ones.[131]

The state-of-the-art framework for fitting the PARAFAC2 model uses the Alternating Least Squares (ALS).[60] This corresponds to optimizing a least-squares criterion, in an alternating fashion, where we solve for a subset of factors by fixing all others. Due to the sparse nature of our EHR input data, we use the algorithm proposed by Perros et al., which also follows the ALS framework, but is optimized for sparse datasets.[51]

**Automatic identification of the number of phenotypes**

Up to now we assume the number of phenotypes $R$ are given but in practice it is unknown and should be learned by the algorithm. Next we describe an extension to [51] for automatically determining R. [190]

Consider that we have computed the factor matrices $\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}$ of the CP model with $R$ components for a tensor $\mathcal{X} \in \mathbb{R}^{I\times J\times K}$. A simple and intuitive way of assessing the quality of the CP model is described in [145, 191]: assuming that $\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}$ are the factor matrices of a Tucker model for $\mathcal{X}$, retrieve the corresponding core tensor $\mathcal{G}$. As mentioned above, the CP is a restricted version of the Tucker model, where the core tensor $\mathcal{G}$ is assumed to be super-diagonal. The core consistency diagnostic (CORCONDIA) [145, 191] essentially assesses the validity of this assumption, by examining whether there are significant deviations from a super-diagonal core tensor. This would imply that the decomposition is somehow flawed (either the pursued rank is not appropriate or the model cannot describe

the data well enough).

Thus, the least-squares problem to retrieve the core tensor $\mathcal{G}$ can be posed as [192]:

$$\min_{\mathcal{G}} ||vec(\mathcal{X}) - (\boldsymbol{U} \otimes \boldsymbol{V} \otimes \boldsymbol{W})\, vec(\mathcal{G})||_F^2 \tag{5.2}$$

with solution: $vec(\mathcal{G}) = (\boldsymbol{U} \otimes \boldsymbol{V} \otimes \boldsymbol{W})^{\dagger}\, vec(\mathcal{X})$, where $\otimes$ stands for the Kronecker product and $\dagger$ stands for the matrix pseudoinverse. Then, if $\mathcal{T}$ is defined as a super-diagonal tensor of ones, the percentage of core consistency is defined as:

$$1 - \frac{\sum_{p=1}^{R}\sum_{q=1}^{R}\sum_{r=1}^{R}(\mathcal{G}(p,q,r) - \mathcal{T}(p,q,r))^2}{R} \tag{5.3}$$

Computing the CORCONDIA measure for CP models is useful for PARAFAC2, because the CP decomposition arises as an intermediate step of the workhorse PARAFAC2 fitting algorithm. [59, 51] Thus, a practical way of estimating the validity of PARAFAC2 model is to estimate the validity of its intermediate CP decomposition which provides a subset of the resulting PARAFAC2 model factors. We refer to [190] for further details of this process.

We summarize the process of extracting the most appropriate model as follows: we define a range to search for the number of phenotypes (in the experiments, we have used $R = \{2, 3, \ldots, 7\}$). For each one of the candidate values of R, we run the PARAFAC2 fitting algorithm 20 times with random initialization of factors, and also with the Singular Value Decomposition (SVD)-based initialization suggested in [193]. This step increases our chances of avoiding local minima. Then, for each candidate target rank R, we choose the solution with the minimum cost function error. [145] Finally, we choose the solution with the highest possible target rank, which still provides a well-specified model with respect to the core consistency diagnostic. The reasoning is that models with larger target rank explain more of the input data variation (i.e., lower approximation error), so we prefer

them, given the fact that the model is well-specified w.r.t. the CORCONDIA measure. We use the standard threshold of $90\%$ to define a well-specified model. [171]

**Identification of patient groups with similar phenotypic behavior**

Since the diagonal $\mathbf{S}_k$ matrix provides personalized phenotype indicators based on the extracted phenotypes, it can be used to extract meaningful groups of patients with similar phenotypic behavior. To do so, we build a patient-by-phenotypes matrix $\mathbf{W} \in \mathbb{R}^{K \times R}$, where the $k$-th row $\mathbf{W}(k,)$ contains the diagonal of $\mathbf{S}_k$. The resulting matrix is essentially a low-dimensional representation of patients; a key property though is that all the patient representations are directly interpretable based on the extracted phenotypes. To visually identify clusters of users with similar utilization profiles, we use the well-known tSNE [194] software, which can reduce the R-dimensional vectors to the 2-dimensional space.

Table 5.1: Summary of dataset statistics. We consider that the k-th patient exhibits a certain phenotype (e.g., Gastrointestinal disorders) based on the coordinate of the diagonal $\boldsymbol{S}_k$ (phenotype indication output vector) with the maximum value.

| **Summary statistics** | **value** |
|---|---|
| Patient count | 1,045 |
| Median and interquartile range (25th-75th percentile) of encounters per patient | 52 (45 - 64) |
| Median and interquartile range (25th-75th percentile) of clinical features per patient | 86 (64 - 109) |
| Median and interquartile range (25th-75th percentile) of event occurrences per patient | 406 (286 - 570) |
| | |
| % of patients sharing Gastrointestinal disorders | 0.4526 |
| % of patients sharing Oncological conditions | 0.2555 |
| % of patients sharing Blood-related disorders | 0.156 |
| % of patients sharing Neurological system disorders | 0.1359 |

## 5.4 Results

### Cohort Selection & Data Extraction

We focus on phenotyping medically complex pediatric population. We model the last 2 years of records of each patient, so that the most recent clinical status is considered. For

example, assume that the most recent record of a patient occurred at day x. We consider all the medical events occurring within the [x-730, x] day range (2 years).

Formally, we define medically complex patients (MCP) as the patients who have Clinical Risk Group (CRG) 5b, number of specialties 3, and total hospital charge $\geq \$31,027$ (ranked as the top 5% annual total charge in the cohort) within 1-year windows. Out of all MCP patients, we considered the ones having at least 40 encounters during the last 2 years of their records. This step ensures that we can study patients exhibiting some form of temporal clinical feature evolution. This led to 1,045 pediatric patients with a total of 59,948 encounters. The maximum number of encounters per patient is 202 and the mean is 57.4. In Table 5.1, we summarize the dataset statistics and the phenotype distribution among patients.

We utilize the diagnoses, medications and procedures as the clinical features for each patient. To enhance interpretability and clinical meaningfulness, we aggregate all the individual medical codes to clinical categories. Their International Classification of Diseases (ICD9) codes are summarized according to the flat (single-level) categorization of Clinical Classification Software (CCS). [195] The provided Current Procedural Terminology (CPT) [196] procedure codes are also summarized using the corresponding CCS categorization for procedure coding. [195] Medications are provided in the internal form of categorization followed by the data provider, so no transformation was needed. As a result of the summarization described, the number of diagnostic categories is 249, the number of medication categories is 455 and the number of procedure categories is 156. Thus, the total number of clinical features considered is 862.

We also filtered the CCS diagnostic and procedure categories, by removing the ones that were too abstract or fragmented to provide direct clinical information, in accordance with the medical experts guidelines. Thus, we removed the diagnostic categories: Residual codes; unclassified and Other aftercare. We also removed the procedure categories: Other diagnostic procedures (interview; evaluation; consultation) and Other therapeutic proce-

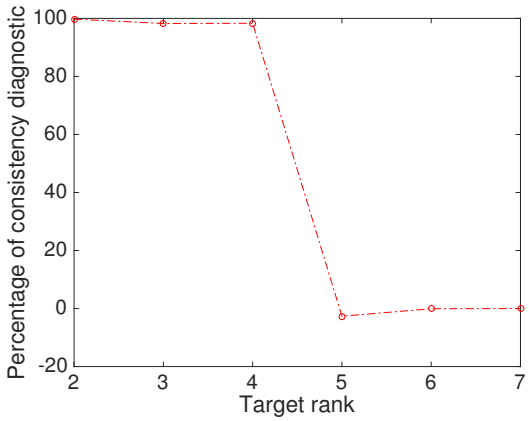



Figure 5.3: Plotting the percentage of consistency diagnostic. We choose the solution providing a well-specified model reflected by the consistency diagnostic. Thus, 4 phenotypes are selected.

dures.

**Study Design**

We used the Matlab implementation of the PARAFAC2 algorithm designed for sparse input, provided in [51]. We also utilized the core consistency diagnostic provided in the N-way toolbox. [157] To increase our chances of avoiding local minima, for each target rank (i.e., R phenotypes pursued), we run the PARAFAC2 fitting algorithm 20 times with random initialization of factors, and also with the Singular Value Decomposition (SVD)-based initialization suggested in [154]. Then, we choose the solution with the minimum cost function error. [145]

**Choosing the number of phenotypes automatically**

In Figure 5.3, we provide both the percentage of consistency diagnostic for various target ranks (i.e., phenotypes). As explained in the Materials and Methods Section, we pick the solution with the highest target rank which still provides a well-specified model (i.e., consistency diagnostic above $90\%$). [171] Thus, for our experiment, the solution with 4 phenotypes is considered as the most suitable. The fact that there is a clear-cut among the CORCONDIA values for solutions with $R \leq 4$ (all above $90\%$) and those with $R \geq 5$ (all close to $0\%$) further boosts our confidence on the appropriateness of the solution

picked. [171]

**Discovery of phenotypes & temporal trends**

First, we provide a clinical definition of each phenotype annotation, as provided by a medical expert. This will aid in introducing the reader to the characteristics of each phenotype and the specificity of the discovered contents to their annotation. Note that the phenotype annotations were provided by a medical expert (Searles) after observing the associated clinical features of each phenotype described in the output model ($\boldsymbol{V}$ matrix in particular).

1. Gastrointestinal disorders are disorders involving the gastrointestinal tract, namely the esophagus, stomach, small intestine, large intestine and rectum, and the accessory organs of digestion, such as the liver, gallbladder, and pancreas. These disorders occur when the gastrointestinal tract does not function properly and may occur with or without an underlying disease.

2. Oncological conditions refers to medical conditions resulting from tumors. There is an extensive number of types of tumors and types of care for tumors.

3. Blood-related disorders are disorders, possibly diseases, of the blood, involving the red blood cells, white blood cells (leukocytes), or platelets (thrombocytes); the tissues in which these elements are formed such as the bone marrow, lymph nodes, and spleen; or of bleeding and blood clotting.

4. Neurological system disorders are disorders of the nervous system. Structural, biochemical or electrical abnormalities in the brain, spinal cord or other nerves can result in a range of symptoms.

In Tables 5.2, 5.3, we provide the detailed phenotype definitions discovered by PARAFAC2. The phenotype contents are ranked according to the value provided by the model for the specific medical feature ($\boldsymbol{V}(i, r)$ value for the $i$-th feature expression as part of the $r$-th phenotype). Thus, we expect that the higher-ranked medical categories of each phenotype

Table 5.2: Gastrointestinal disorders & Oncological conditions phenotype definitions extracted by PARAFAC2.

| **Gastrointestinal disorders** | **Oncological conditions** |
|---|---|
| MED_Analgesic/antipyretics non-salicylate | MED_Heparin and related preparations |
| MED_Analgesics narcotics | DIAG_Maintenance chemotherapy; radiotherapy |
| MED_Sodium/saline preparations | PROC_Cancer chemotherapy |
| MED_Antihistamines - 1st generation | MED_Antiemetic/antivertigo agents |
| MED_Antiemetic/antivertigo agents | DIAG_Leukemias |
| MED_Potassium replacement | DIAG_Immunity disorders |
| MED_Sedative-hypnotics non-barbiturate | MED_Sodium/saline preparations |
| MED_Topical local anesthetics | MED_Topical local anesthetics |
| MED_Laxatives and cathartics | MED_Antineoplastic - antimetabolites |
| MED_Heparin and related preparations | MED_Antihistamines - 1st generation |
| MED_Analgesics narcotic anesthetic adjunct agents | MED_General anesthetics injectable |
| MED_Beta-adrenergic agents | DIAG_Cancer of brain and nervous system |
| MED_Absorbable sulfonamide antibacterial agents | PROC_Laboratory - Chemistry and hematology |
| PROC_Microscopic examination (bacterial smear; culture; toxicology) | MED_Analgesics narcotic anesthetic adjunct agents |
| DIAG_Other gastrointestinal disorders | MED_Glucocorticoids |
| MED_Iv solutions: dextrose-saline | MED_Iv solutions: dextrose-saline |
| MED_General anesthetics injectable | |
| DIAG_Other nutritional; endocrine; and metabolic disorders | |
| PROC_Other laboratory | |
| MED_Topical preparations,antibacterials | |

are the ones exhibiting a higher recording frequency among the listed contents. We truncate the medical categories with values lower than $0.05$ and list up to $20$ categories for each phenotype. Overall, the listed phenotype definitions were endorsed by a medical expert as homogeneous and descriptive of the corresponding phenotype annotation. A notable characteristic of the phenotype definitions extracted is that neurological system disorders are mainly characterized by a multiple related diagnoses and procedures, while gastrointestinal disorders and oncological conditions are mainly characterized by medications. Apart from different pathophysiological characteristics, this indicates that the phenotype definitions vary w.r.t. the most frequently used modality as the means of treatment (medications for gastrointestinal disorders and oncological conditions, as compared to procedures for neurological system disorders).

In Figure 5.4, we are showcasing a representative example of a patient temporal signature alongside with the raw EHR for this patient. The temporal signature produced by PARAFAC2 provides a succinct summary of the patients phenotypic temporal trends: as can be verified by the raw EHR data, the temporal signature successfully captures the period when cancer treatment is underway.

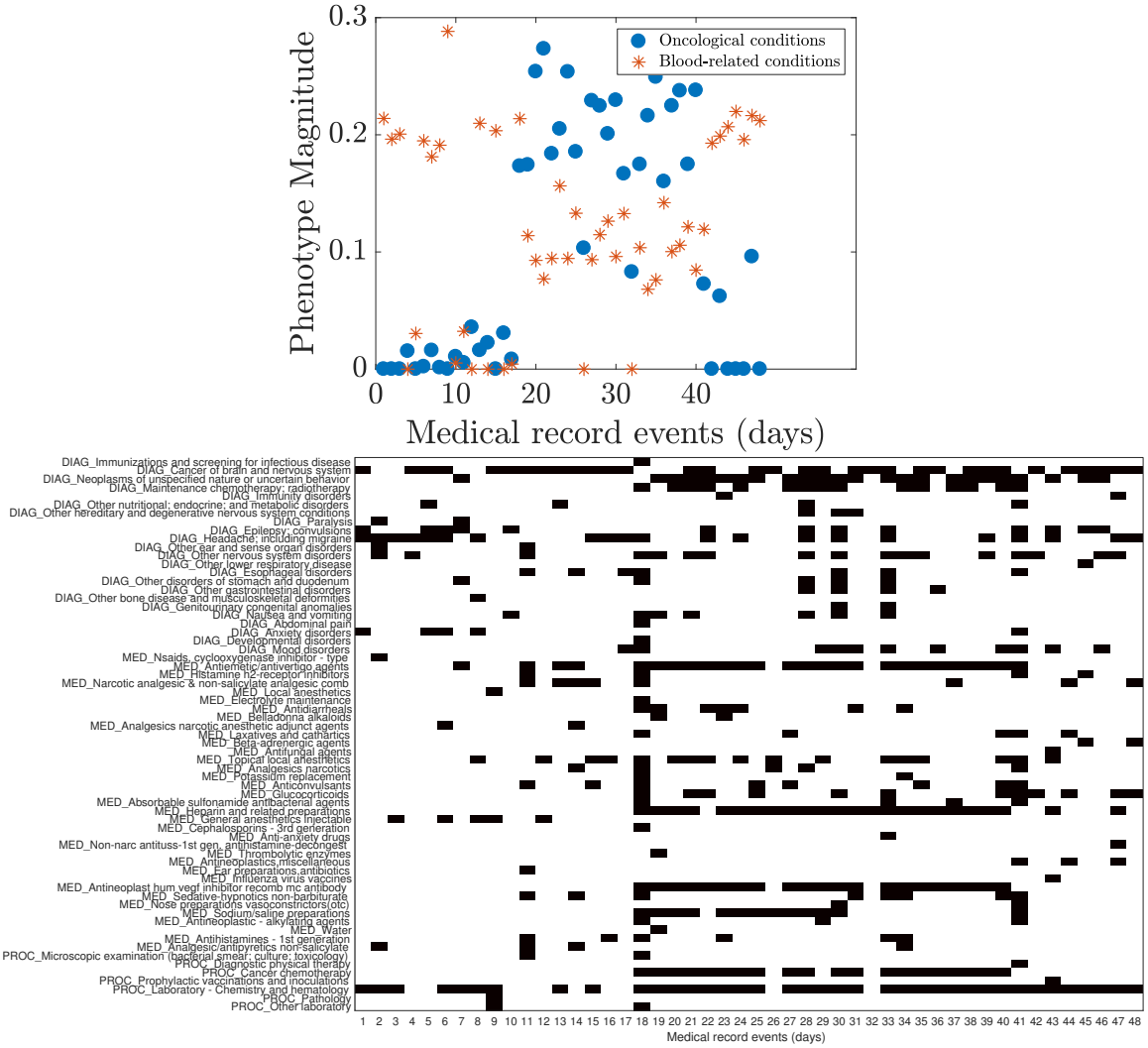



Figure 5.4: **Upper part:** Temporal signature of phenotypes for a certain patient. **Lower part:** Raw EHR information in the form of clinical categories, for the same patient. PARAFAC2 successfully captures the period of the patient's history where cancer treatment is underway. It also captures the blood-related disorders' phenotype due to the presence of hematology lab tests throughout her history.

Table 5.3: Blood-related conditions & Neurological system disorders phenotype definitions extracted by PARAFAC2.

| **Blood-related disorders** | **Neurological system disorders** |
|---|---|
| PROC_Laboratory - Chemistry and hematology | PROC_Physical therapy exercises; manipulation; and other procedures |
| PROC_Other laboratory | DIAG_Rehabilitation care; fitting of prostheses; and adjustment of devices |
| PROC_Microscopic examination (bacterial smear; culture; toxicology) | DIAG_Other nervous system disorders |
| DIAG_Immunity disorders | DIAG_Other connective tissue disease |
| DIAG_Chronic kidney disease | PROC_Other physical therapy and rehabilitation |
| DIAG_Deficiency and other anemia | DIAG_Other nutritional; endocrine; and metabolic disorders |
| DIAG_Other nutritional; endocrine; and metabolic disorders | DIAG_Paralysis |
| MED_Heparin and related preparations | DIAG_Developmental disorders |
| DIAG_Other liver diseases | DIAG_Other gastrointestinal disorders |
| DIAG_Complication of device; implant or graft | DIAG_Epilepsy; convulsions |
| MED_Immunosuppressives | PROC_Diagnostic physical therapy |
| MED_Sodium/saline preparations | |
| MED_Analgesic/antipyretics non-salicylate | |
| DIAG_Sickle cell anemia | |
| PROC_Nonoperative urinary system measurements | |

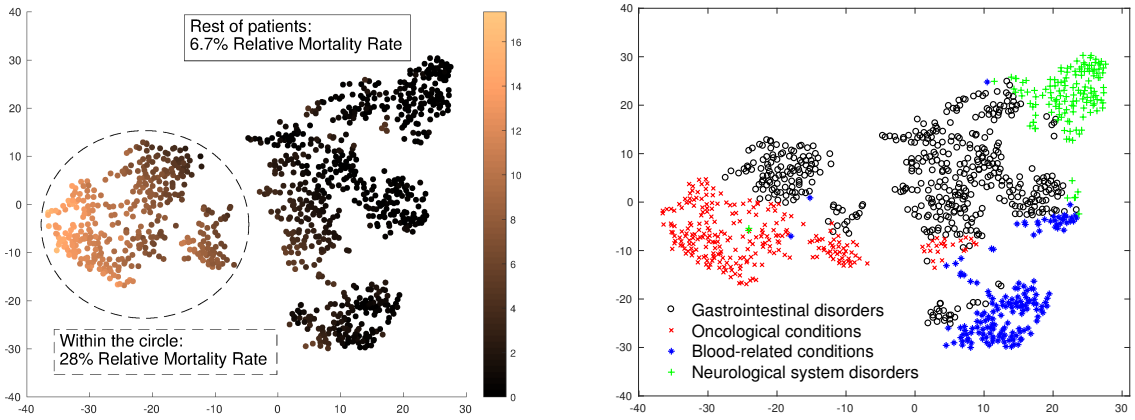



Figure 5.5: tSNE visualization of patient representations learned by PARAFAC2. Each point corresponds to a certain patient, mapped to the 2D space. **Left Part:** The color of each point (patient) corresponds to the intensity of the "Oncological conditions" phenotype. $28\%$ of the $375$ patients belonging to the circled area are recorded to have deceased. On the other hand, only $6.7\%$ of the $670$ rest of patients are reported to have deceased. **Right Part:** The color and marker type for each point is decided based on the phenotype (vector coordinate) with the maximum value for the corresponding patient.

**Identification of higher-risk patient subgroups**

Now we assess whether PARAFAC2 can be used to extract meaningful patient subgroups, by exploiting the learned patient phenotype indicators $\mathbf{S}_k$, as described in the closing subsection of the Materials and Methods section.

We build a patient-by-phenotypes matrix $\boldsymbol{W} \in \mathbb{R}^{K \times R}$, where the k-th row $\boldsymbol{W}(k,:)$ contains the diagonal of $\boldsymbol{S}_k$. The largest variance among the columns of W corresponds to the Oncological conditions phenotype; thus, we attempt to visualize the patient representations and examine whether there is clear distinction between lower and higher weights of this phenotype. To visualize the representations, we used the well-known tSNE [194] software, by reducing the 4-dimensional vectors to the 2-dimensional space. On top of that, we color the marker corresponding to each patient with respect to the weight associated with the Oncological conditions phenotype. In the left part of Figure 3, we present the results of this experiment. The distinction between lower and higher values corresponding to the second phenotype is clear enough, giving rise to two patient sub-groups. For each one of the sub-groups, we calculate the relative mortality rates and identify significant survival variations (28% in the higher-risk group over 6.7% for the rest of patients). Thus, the low-dimensional patient representations of PARAFAC2 are shown to distinguish between higher and lower mortality risk patients. This hints that interpretable low-dimensional patient representations produced by PARAFAC2 may be useful for predictive modeling tasks (e.g., mortality prediction). We elaborate on that in the Discussion Section.

Furthermore, in the right part of Figure 1, we showcase the same tSNE patient representation described above. In this case, we color each patient based on the most-expressed phenotype regarding her low-dimensional representation (for the k-th patient, we pick the coordinate of the $\boldsymbol{W}(k,:)$ vector with the highest value). The patients having maximal expression with respect to all but the Gastrointestinal disorders phenotype are closely clustered together in the tSNE representation.

## 5.5 Discussion

Raw EHR information (even in the form of clinical categories) is often too complex for clinical providers to provide simple and intuitive understanding of patient phenotypes. This is why manual phenotyping through chart review is considered a time-consuming and laborious task. [38, 7] This highlights the importance of computational temporal phenotyping towards health management of medically complex populations. Our proposed approach leads to an interpretable model with uniqueness guarantees which is computed through an algorithm that has been demonstrated to scale to hundreds of thousands of patients. [51]

For the reasons above, we believe that our approach is a promising candidate solution for clinical deployment. The most appropriate use case in clinical practice is through integration in clinical decision support systems (CDSSs). Such systems are expected to lower healthcare costs, improve efficiency and reduce patient inconvenience. [197] In particular, our temporal phenotyping solution could support clinicians decisions, when no clear treatment guidelines exist for their patients. [198, 4, 6] Exploiting EHR data to identify patients with common phenotypes and temporal trends may guide clinicians towards interventions that have been empirically demonstrated to be effective in similar populations. This is particularly important for populations such as the medically-complex children we focus on in this work; due to the complex underlying pathophysiology, many medically-complex patients may not precisely match textbook guidelines derived from gold standard evidence.

An immediate application of our computed phenotypes is towards interpretable predictive modeling. As showcased in the experiments, the learnt low-dimensional patient representations (matrix $\mathbf{S}_k$) can be utilized to identify significant mortality rate variations. Thus, feeding them into a predictive modeling task is a promising research direction. The key benefit of doing so as opposed to deploying a predictive model on the raw, noisy features is improved interpretability: the features of the predictive model would be the $R$ validated phenotypes, thus interpreting the contribution of each one of them w.r.t. the out-

come measure (e.g., mortality) is a much simpler process than considering all the possibly noisy input features recorded. Previous literature [24] has demonstrated success on a similar methodology using the simpler PARAFAC tensor model, which cannot directly handle the longitudinal nature of EHR data. Assessing whether the more flexible PARAFAC2 model used in this work could provide more accurate patient representations for predictive modeling is an immediate next step of this work.

Another potential application of EHR-based computational phenotyping is exploiting the computed phenotype definitions to identify eligible patients for recruitment in clinical trials. [7] This is important as manual chart reviewing is impractical as a means to perform EHR-based patient matching to trials. [199] This is particularly relevant for the medically-complex children cohort we utilized in this study, since there may be multiple patients undergoing unsuccessful treatments who may be benefited by participating in a clinical trial.

A limitation of our approach is that a single medical expert (Searles) validated and annotated the computed phenotype definitions. We plan to strengthen the clinical validation of our results by onboarding more experts.

## 5.6 Conclusion

In this work, we tackle the challenge of temporal phenotyping of a medically complex pediatric population. We report the discovery of four medically complex phenotypes and representative examples of their temporal trends in different patients, which were endorsed by a clinical expert. We also demonstrated the usefulness of exploiting the low-dimensional patient representations produced by PARAFAC2, towards extracting subgroups of patients with significant survival variations.

# CHAPTER 6

# INTERPRETABLE TEMPORAL PHENOTYPING OF PRE-DIAGNOSTIC HEART FAILURE VIA INTEGER-CONSTRAINED PARAFAC2 FACTORIZATION


**Summary**

Can distinct and interpretable pre-diagnostic heart failure (pre-HF) phenotypes be identified from longitudinal EHRs in an efficient manner? In this Chapter, building upon the results of Chapters 3 and 4, we propose the integer-constrained PARAFAC2 and apply that to extract concise, pre-HF phenotype definitions on HF patients from Sutter Palo Alto Medical Foundation. We quantitatively demonstrate the increased interpretability of integer-based phenotypes as compared to the real-valued model, by evaluating both in terms of conciseness and feature overlap across different phenotypes.


Our goal is to extract interpretable phenotypes of pre-diagnostic heart failure (pre-HF) from longitudinal electronic health records (EHRs) in an efficient manner. Identifying the various stages and progress of pre-HF is important due to: a) the prevalence and complications associated with HF; b) the heterogeneity of HF subtypes and c) the potential of developing adaptive treatment strategies based on a better understanding of pre-HF patient progress. The EHRs provide a rich data source that can be used to study the temporal progression of pre-HF. Our aim is to do so in an efficient and interpretable manner which both avoids the need for manual chart review and automatically provides concise and clinically-meaningful groups of EHR features indicating each one of the phenotypes.

We analyze longitudinal EHR data of 2,577 HF patients from Sutter Palo Alto Medical Foundation (Sutter-PAMF). We consider the diagnoses, medications and procedures

recorded within a 2-year observation window ending 6 months before each patient met the operational criteria of HF diagnosis (HFDx). We propose an integer-constrained version of the PARAFAC2 tensor factorization to tackle the pre-HF temporal phenotyping challenge. Through our proposed approach, we extract: a) computational phenotypes as groups of features associated with integer scores indicating the features frequency, b) integer patient representations indicating the level of presence of a phenotype in the corresponding patients records, and c) real-valued factors indicating when a phenotype occurred across the patient history. We also describe how to extract the earliest occurrence of each phenotype across all associated patients; the resulting insights may be impactful towards guiding the design (e.g., length of prediction window) of predictive models based on the underlying data. We guide the choice of the number of phenotypes via a stability-driven criterion, promoting reproducible results when starting from multiple different initialization points.

We identified 16 phenotypes reflecting different clinical manifestations and healthcare processes indicative of pre-HF patients. We quantitatively demonstrate the increased interpretability of the resulting phenotypes based on integer constraints, as compared to the corresponding real-valued model; to do so, we evaluate both in terms of model conciseness and feature overlap across different phenotypes.

## 6.1 Introduction

Heart failure (HF) is a leading cause of morbidity and mortality. The lifetime risk in individuals aged 55 years and older is about one in five and the 5-year survival ranges from 2050% after first diagnosis. [200] At the same time, HF is a highly heterogeneous condition and gaining a concrete understanding of both its clinical characterization as well as its trajectory over time remains an open challenge. [201, 202] Tackling this challenge is important as it may allow for more targeted treatment development. [203]

Electronic health records (EHRs) provide a rich source of readily-available data that can be used to study the temporal progression of HF. Focusing on identifying EHR-based

phenotypes capturing the pre-diagnostic heart failure (pre-HF) progression is important due to the following reasons: a) the phenotypes extracted may provide the foundations for developing clinically actionable protocols of preventive care, and b) they may be further exploited as succinct feature summarizations in downstream predictive tasks which utilize EHR data (e.g., to perform early detection of HF in an interpretable manner).

Manually identifying pre-HF phenotypes through EHR chart review is time-consuming and not scalable to the large patient cohorts available. [38] Computational rule-based approaches [165, 166] suffer from similar issues; they incur substantial burden on both medical informatics and clinical experts in order to develop carefully-crafted feature sets and possible ranges for each individual phenotype. As a result, it is extremely challenging to scale these approaches for large number of patients and heterogeneous cohorts. [167]

Our goal is to extract phenotype candidates (i.e., patient groups exhibiting similar pre-HF subtypes) along with information about the timing of phenotype occurrence across different patients without expert supervision. Our proposed methodology is based on tensor factorization, which is a set of methods extracting the latent correlation structure in diverse and multidimensional input data which can be naturally represented as a tensor (i.e., generalization of matrix for higher orders). [11] Tensor factorization methods have the following desirable properties towards our target task: a) multiple phenotypes can be simultaneously derived from data without user supervision or domain expertise; b) diverse, high-order interactions are captured, such as the ones occurring among patients, medical features and patient encounters; c) concise phenotype criteria are generated as output, simplifying interpretation of complex relationships; d) soft clusters of patients can be derived where patients may have multiple phenotype assignments with different degrees of association.

Tensor factorization has been successfully applied and extended towards unsupervised phenotyping with multiple proof-of-concept demonstrations (e.g., [24, 134, 48]). Recent work based on tensor factorization has additionally tackled the challenge of extracting timing information of phenotype occurrence across different patients, [61, 51, 204] a task

we call as unsupervised temporal phenotyping. [10] Still, the model weights capturing feature associations to phenotypes and phenotype assignments to patients are dense and real-valued. Factor density results in lengthy feature groups which share several features among the phenotype candidates, thus making the model less interpretable (due to the lack of conciseness) and less actionable (due to the limited discriminatory power among different phenotype candidates). Also, when the input data represent event occurrences (as happens in this work), the non-negative factor weights extracted are direct indicators of frequency, reflecting: a) how often does a phenotype occur in a patients records and b) how often does a feature occur as part of a feature group across all associated patients. However, real-valued factor elements may not be the most intuitive indicators of frequency-based relationships and integer scores on a standardized scale may be more appropriate to describe relative frequency relationships among factor elements.

To tackle the challenges described above, we propose the integer-constrained PARAFAC2 tensor factorization. Our proposed model inherits favorable properties from its real-valued counterpart [58] for temporal phenotyping purposes: [61] a) it preserves model uniqueness under mild assumptions, [59] thus avoiding rotational model ambiguities which would hinder interpretability, and b) it can naturally handle core properties of EHR data, such as sparsity and variability of the number of encounters across different patients. Through our approach, we extract: a) computational phenotypes as groups of features associated with integer scores indicating the features frequency, b) integer patient representations indicating the level of presence of a phenotype in the corresponding patients records, and c) real-valued temporal signatures indicating the evolution of those phenotypes over time for each patient. We also describe how to extract the time segment of earliest occurrence of each phenotype across all associated patients. We guide the choice of the number of phenotypes via a stability-driven criterion, promoting reproducible results when starting from different initial points.

To automatically identify pre-HF phenotypes, we consider 2,577 HF patients from Sut-

ter Palo Alto Medical Foundation (Sutter-PAMF) and a 2-year observation window ending 6 months before each patient met the operational criteria of HF diagnosis (HFDx). [205] We identified 16 clinically-meaningful phenotypes reflecting different clinical manifestations and healthcare processes indicative of pre-HF. Apart from the phenotypes clinical validation, we empirically assessed their interpretability by using sparsity and feature overlap among different phenotypes as proxy measures. Through this experiment, we demonstrate that our proposed approach achieves highly-interpretable solutions, as the integer constraints imposed promote sparser and less overlapping phenotypes than the real-valued model.

## 6.2 Related Work

**Heart failure phenotyping** There are several prior works investigating risk factors, signs and symptoms derived based on EHR data analysis. Vijayakrishnan et al. examine the prevalence of Framingham criteria in both HF case and control patients through unstructured clinical notes processing. [206] Shah et al. employ hierarchical clustering to distinguish between different subtypes of heart failure with preserved ejection fraction (HFpEF). [203] Uijl et al. investigate age- and sex-specific differences in prevalence among incident HF patients. [200] Knorek et al. examine the use of loop diuretics as a clinical predictor of HF in the period preceding its formal diagnosis. [207] Finally, there is a line of supervised phenotyping approaches which tackle the early HF prediction challenge and investigate several design choices related to this task, such as the machine learning method to utilize and the minimum amount of training data needed. [205, 208, 110].

Overall, none of the aforementioned works automatically extracts interpretable feature ensembles which can characterize a specific HF phenotype from EHR data without access to gold-standard labels; they also do not focus on characterizing the temporal evolution of pre-HF phenotypes.

**Unsupervised temporal phenotyping** Several prior works have tackled the challenge of

unsupervised temporal phenotyping via a wide range of approaches, which we summarize below. The most relevant ones to our work apply and extend the real-valued PARAFAC2 factorization for phenotyping purposes; [61, 51, 204] as we demonstrate in the experiments, the phenotypes derived from the real-valued PARAFAC2 model are harder to interpret and potentially less actionable than the ones derived from the proposed integer-constrained counterpart.

Another category of approaches follows factorization-based temporal phenotyping. [209, 139, 177, 178] However, these works do not provide model uniqueness guarantees. As a result, there may exist many arbitrarily rotated versions of a solution yielding the same approximation error and it is unclear to the practitioner which one of those solutions provides the true latent factors.

Another approach to factorization-based phenotyping is to flatten the temporal mode by aggregating clinical features over time so that different patients can be aligned for analysis purposes. [24, 134, 48, 135, 210, 179, 126, 211, 212] However, such a pre-processing ignores the temporal ordering of clinical events which is necessary to understand phenotype progression.

Embedding-based approaches have also appeared in the recent literature which extend the idea of word embeddings [185] to extract concept and visit [186], care event [213] and patient [167] representations. Those works do not focus on extracting phenotype trajectories over time. Simpler clustering-based approaches have also been proposed [182], which only provide with the patient clusters, without the corresponding phenotype definitions. In the absence of those definitions, both tasks of validating and generalizing the result to different populations are challenging. Simpler clustering approaches may additionally require pre-processing of input longitudinal data into segments so that those are comparable across different patients. [183, 184].

Several probabilistic approaches have been proposed as well. [173, 175] focus on modeling and forecasting continuous variables such as vital signs or uric acid concentration

over time. As such, they are not tailored to handle the sparsity and irregularities present in EHR high-dimensional discrete event sequences. In [8], the authors use a Markov Jump Process to model the stage transitions of a certain disease. Unlike our proposed method, their approach requires a pre-defined set of disease stages to be set (apart from the target number of phenotypes and other additional input parameters). They also report practical obstacles, such as the need to pre-process the input data into time windows to deal with input EHR data sparsity, as well as the requirement to specify anchor features as a form of supervision for every phenotype in order to achieve interpretable results. Neither of those requirements is necessary for our method. In [174], the authors focus on extracting temporal trajectories of individual input variables, rather than phenotypic trajectories, which capture the evolution of clinically-meaningful groups of features. Static probabilistic phenotyping approaches have also been proposed (such as the UPhenome model [176]) which fail to encode any temporality about the patient records.

Sequential pattern mining approaches have also appeared which capture a certain flow of care provided based on individual features, rather than phenotype definitions based on feature groups which evolve over time. [10, 187] Such pattern mining approaches also fail to provide succinct computable patient representations which can be further utilized in downstream predictive tasks. Finally, association analysis of structured codes and visualization of the resulting patterns has also been pursued. [214, 188, 215, 216] Those approaches only provide a high-level understanding of code associations over time and do not enable the extraction of precise phenotype definitions.

## 6.3 Materials and Methods

**Data description** We use data from Sutter Palo Alto Medical Foundation (Sutter-PAMF) primary care patients. Sutter-PAMF has been using an Epic Systems Corporation EHR for more than a decade. The EHR dataset contains documentation of care delivered in the outpatient setting. We focus on extracting structured code information recorded as part of each

encounter, medication order and procedure order for each patient. We utilize the following types of available data in the EHR: the International Classification of Diseases (ICD-9) diagnostic codes recorded for each encounter, [43] the therapeutic subclass information in the form of the Anatomical Therapeutic Chemical Classification System (ATCCS) [112] provided as part of the medication orders, and the Current Procedural Terminology (CPT) codes provided as part of the procedure orders. For every medication and procedure order, we utilize the target diagnosis information available and include the corresponding ICD-9 codes as part of the input data for each patient. Finally, since we target longitudinal modeling, we make use of the date information for each recorded code. For this purpose, we timestamp the codes recorded using the encounters contact date. In case that such date information is not available (e.g., Pharmacy medication orders may not have a matching encounter identifier), we use the order date information as timestamp.

**Heart failure case patient definition** We use the following operational criteria to define a HF diagnosis (HFDx): [206]

1. Qualifying ICD-9 codes for HF appeared as an encounter diagnosis or as the indication for a medication order.

2. A minimum of three clinical encounters with qualifying diagnostic codes had to be present within a year of each other. The HFDx date is considered to be the earliest of those dates.

**Input data formulation** Each patient's records are represented through a binary matrix $\mathbf{X_k} \in \{0,\ 1\}^{I_k \times J}$. Rows correspond to a sorted sequence of dates, in increasing order from earliest to latest, on which a clinical event for the $k$-th patient was recorded. Columns correspond to clinical features (illustrated in Figure 6.1)). $\mathbf{X_k}\,(i,\ j) = 1$ corresponds to the occurrence of the j-th feature on the i-th sequenced date of the k-th patient's records. Cells containing a value of zero reflect the absence of the corresponding event. Duplicate feature occurrence (e.g., when a diagnostic code is recorded both as part of the encounters as well

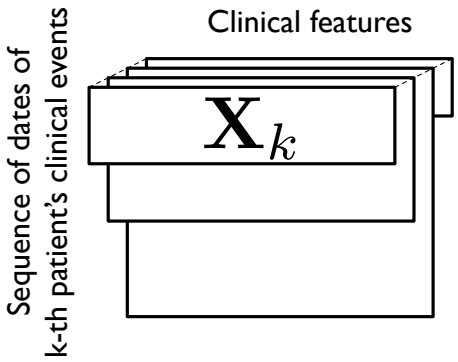



Figure 6.1: Input for our proposed approach: each patient's records are represented through a matrix where rows correspond to a sorted sequence of dates (from earliest to most recent) on which a clinical event occurred for the k-th patient and columns correspond to the clinical features monitored (common across all patients). Each one of those matrices is binary: $\mathbf{X}_k(i, j) = 1$ corresponds to the occurrence of the j-th feature on the i-th sequenced date of the k-th patients records. Cells containing a value of zero reflect the absence of the corresponding event.

as a target diagnosis in a medication order) is ignored.

**Proposed model and interpretation** We propose an integer-constrained version of the PARAFAC2 model. [58] Considering $\mathbf{X_k} \in \{0,\ 1\}^{I_k \times J}$,$k = \{1,\ \ldots,\ K\}$ as input the model takes the following form:

$$\mathbf{X_k} \approx \mathbf{U_k}\ \mathbf{S_k}\ \mathbf{V^T}$$

where $\mathbf{U_k} \in \mathbb{R}^{I_k \times R}$, $\mathbf{S_k} \in \{0,\ \ldots,\ \tau\}^{R \times R}$ , $\mathbf{V} \in \{0,\ \ldots,\ \tau\}^{J \times R}$, $\mathbf{S_k}$ is diagonal and $\mathbf{V^T}$ is the transpose of $\mathbf{V}$ and $\tau$ is a small positive integer defined by the user. We illustrate the output model factors in Figure 6.2. We define their interpretation as follows:

- $\mathbf{V}$ is the **phenotype definition** factor and is common across all $K$ patients: the non-zero elements of thej-th column $\mathbf{V}(:,\ j)$ reveal the clinical features associated with the $j$-th phenotype. The integer constraints imposed on the feature weights enable interpreting them as distinct levels of frequency. For example, in Figure 6.2, "Essential hypertension" is expected to be 5 times more frequent than the rest of features with non-zero weights (e.g., "ACE Inhibitors") in the records of patients exhibiting

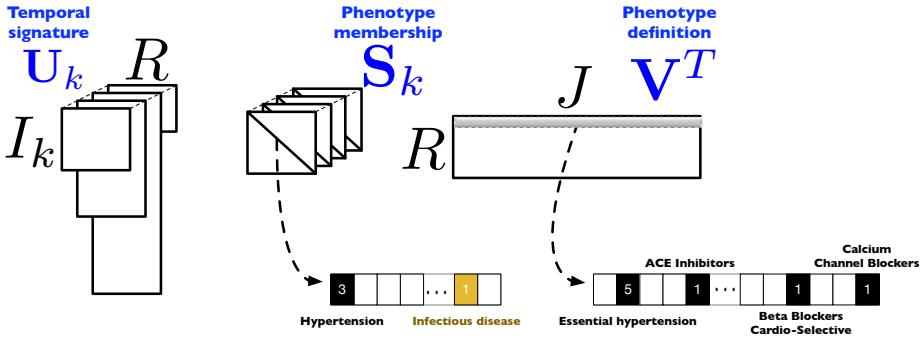



Figure 6.2: Illustration of the proposed integer-constrained PARAFAC2 model.

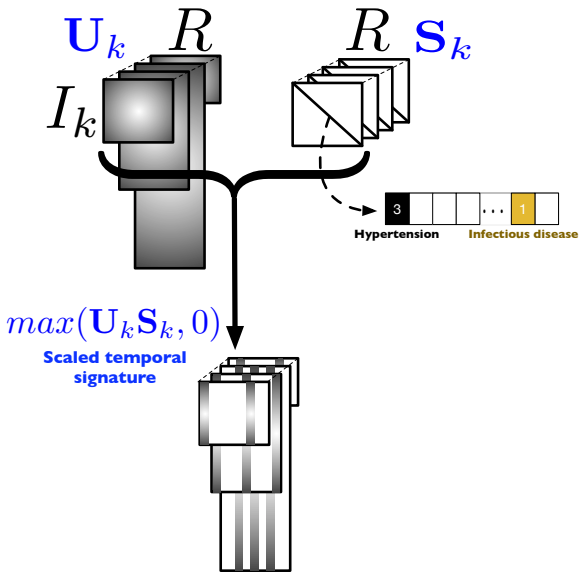



Figure 6.3: Scaled temporal signature extraction for each patient. Note that multiplying each $\mathbf{U}_k$ with $\mathbf{S}_k$ re-scales and sparsifies the k-th temporal signature, due to the implicit sparsity achieved through the integer constraints on $\mathbf{S}_k$.

this specific phenotype.

- $\mathbf{S_k}$ is the **phenotype membership** factor. Its non-zero elements along the diagonal reveal the phenotypes expressed by the k-th patient. Similarly to the interpretation of the phenotype definition factor, we can interpret integer weights as distinct levels of phenotype frequency. For example, in Figure 6.2, the "Hypertension" phenotype is expected to be 3 times more prevalent than the "Infectious disease" one in the k-th patient's records.

Apart from permitting a frequency-based interpretation, the integer constraints imposed on the factors above implicitly achieve sparse solutions, further promoting interpretability.

- $\mathbf{U_k}$ corresponds to the **temporal signature** ; a positive element $\mathbf{U_k}(i,\ j)$ reflects that phenotype $j$ was expressed during the $i$-th sequenced date of records of the $k$-th patient. We also define the **scaled temporal signature** as$\mathbf{U_k'}\ := \max(\mathbf{U_k}\ \mathbf{S_k}$, 0), where $max$ is taken element-wise (Figure 6.3). The benefit of considering such a scaled version of $\mathbf{U_k}$ lies in scaling each $\mathbf{U_k}(:,\ j)$ column with the appropriate integer phenotype membership weight $\mathbf{S_k}(j,\ j)$. Importantly, due to the implicit sparsity achieved by representing $\mathbf{S_k}$ on a small integer scale ($\{0,\ \ldots,\ \tau\}$), several columns of the resulting $\mathbf{U_k}\ \mathbf{S_k}$ product will be completely zero, thus enhancing interpretability. Finally, due to the fact that both the input as well as the phenotypes' definition factor $\mathbf{V}$ are non-negative, we only consider the positive elements of the $\mathbf{U_k}\ \mathbf{S_k}$.

**Identifying earliest phenotype occurrence** In this Section, we describe how to identify the earliest time segment of occurrence of a pre-HF phenotype across all associated patients. For example, we enable detecting which percentage of patients exhibit the hypertension phenotype up to 2 years before the HF diagnosis date. This information may be useful in guiding the design of predictive models based on the underlying data (e.g., by considering

a prediction window such that phenotypes corresponding to well-known risk factors of HF are captured for most associated patients).

We consider a split of our observation window into $b$ time segments. Those do not have to span an equal period of time and can vary depending on the disease or cohort under consideration. For the needs of studying pre-HF phenotypes, we use $b = 4$ segments of 6 months each. Then, we consider the scaled temporal signature factor $\mathbf{U}'_{\mathbf{k}} := \max(\mathbf{U_k}\,\mathbf{S_k}, 0)$, and normalize each $j$-th column by the maximum column value such that $\mathbf{U_k}\,(:,\,j) := \mathbf{U}'_{\mathbf{k}}\,(:,\,j)\,./max(\mathbf{U}'_{\mathbf{k}}\,(:,\,j))$ where ./ denotes elementwise division. Then, we consider as earliest time segment of occurrence of phenotype $j$ in the records of $k$-th patient as the minimum index $l$ for which $\mathbf{U_k}\,(i,\,j) > \rho,\;\; i \in B_{k,l}$ where $B_{k,l}$ is a set storing the indices of $k$-th patient corresponding to the $l$-th time segment and $\rho$ is a user-defined threshold (we empirically choose that to be equal to 0.5).

**Proposed fitting algorithm** The workhorse framework fitting the PARAFAC2 model is the Alternating Least Squares (ALS). [59] This framework dictates optimizing a least squares criterion in an alternating fashion, i.e., fixing a set of factors and solving for all others. The objective is as follows:

$$\min_{\{\boldsymbol{U}_k\},\{\boldsymbol{S}_k\},\boldsymbol{V}} \sum_{k=1}^{K} \left|\left|\mathbf{X_k} - \mathbf{U_k}\mathbf{S_k}\mathbf{V^T}\right|\right|_F^2$$

subject to $\mathbf{S_k}$ being diagonal. In order to guarantee model uniqueness under mild assumptions, Harhsman imposed a cross-product invariance constraint which dictates that $\mathbf{U_k^T}\mathbf{U_k}$ is invariant $\forall k \in \{1,\,\ldots,\,K\}$.. [58] In the context of our phenotyping application, this implies that the correlations among different phenotypes are kept constant, across all patients. Uniqueness ensures that a factorization is pursuing the true latent factors, rather than an arbitrary rotation of them. In the context of phenotyping, using unique models provide with greater license to interpret their output factors as indicative of the true underlying phenotypes. [14, 20]

To satisfy the cross-product invariance constraint, Harhsman imposed that:

$$\mathbf{U_k} = \mathbf{Q_k H},\ \mathbf{Q_k^T Q_k} = \mathbf{I}$$

Thus, the following ALS framework can be employed which iteratively solves for 2 sub-problems until a convergence criterion is met:

1. Solving for $\mathbf{Q_k}$, by fixing the rest of factors, under the constraint that $\mathbf{Q_k^T Q_k} = \mathbf{I}$. The problem reduces to an Orthogonal Procrustes problem and can be optimally solved through the Singular Value Decomposition. [109]

2. Solving for H $\mathbf{S_k}$ $\mathbf{V^T}$ with $\mathbf{Q_k}$ fixed. This step reduces to a CP factorization of a tensor (i.e., multi-way array) $\mathcal{Y} \in \mathbb{R}^{\mathbf{R} \times \mathbf{J} \times \mathbf{K}}$, whose k-th frontal slice is $\mathbf{Y}_k = \mathbf{Q_k^T X_k}$, where $\mathbf{Y}_k \in \mathbb{R}^{\mathbf{R} \times \mathbf{J}}$. [59] In order to impose integer constraints on $\mathbf{S_k}$, $\mathbf{V}$, we solve this step through the recently-proposed SUSTain approach, which performs integer tensor factorization. [48] Also, in order to exploit the sparse nature of input data, we solve the bottleneck MTTKRP step of CP factorization through the SPARTan approach. [51]

**Choosing the number of phenotypes** We use the stability-driven criterion proposed in [117] in order to guide the choice of the target rank $R$ (i.e., the number of phenotypes). Intuitively, this criterion promotes a target rank for which several runs with different initial points return reproducible factors. We choose the phenotypes factor $\mathbf{V}$ as the factor under assessment.

### 6.4 Results

**Data processing** The observation window considered for each patient is [-2.5 years, -6 months] before the HFDx date. A total of 69,817 unique encounters were processed.

A standard step towards improving interpretability and clinical meaningfulness is to transform the codes occurring in the patient records to clinical categories. We transform

the ICD-9 diagnostic codes to hierarchical Clinical Classifications Software (CCS) level-3 codes. [195] We omit the diagnostic category "Dx: Residual codes; unclassified; all E codes [259. and 260.]" since it does not represent a concrete clinical category, but an aggregation of residual diagnostic codes. We also group the normalized drug names (i.e., combining all branded names and the generic name for a medication) based on therapeutic subclasses using the Anatomical Therapeutic Chemical Classification System (ATCCS). [112] We transform the CPT codes available to CCS procedure categories. [195]

Based on the data extraction described above, our initial cohort consisted of 3,362 patients described by 836 categories. We excluded clinical categories appearing in less than 1% of patients; thus, we excluded categories appearing in less than 34 patients. The number of excluded categories was 481. Due to filtering features, we had 4 patients left with empty records, which we removed.

Since we are interested in studying the longitudinal nature of pre-HF phenotypes, we also omitted patients having less than 10 distinct dates of events ($I_k < 10$). [205] 781 patients were omitted through this step.

We present summary statistics describing our input data in Table 6.1

Table 6.1: Summary statistics.

| **Summary statistics** | **Value** |
|---|---|
| Number of patients | 2,577 |
| Number of features | 355 |
| Number of diagnostic categories | 178 |
| Number of medication subclasses | 119 |
| Number of procedure categories | 58 |
| Number of event occurrences | 257,013 |
| Median (25th - 75th percentile) of distinct dates per patient | 27 (17 - 44) |
| Median (25th - 75th percentile) of clinical features per patient | 29 (18 - 42) |
| Median (25th - 75th percentile) of event occurrences per patient | 75 (42 - 135) |

**Study design** We used Matlab R2017b for the implementation, and utilized several functionalities based on SPARTan [51] and SUSTain [48] methods. [217, 218] We also accredit the N-way Toolbox [157] and the Tensor Toolbox [114], from where we have adapted some

functionalities. We plan to open-source our implementations and make them publicly available post-acceptance.

We empirically observed that initializing the integer factors with a rounded version of the real-valued model is less susceptible to local minima than random integer initialization. Naively rounding real-valued solutions usually leads to poor model fit, due to blindly collapsing all values $< \ 0.5$ to $0$. We have designed a more sophisticated rounding approach which first performs scaling to avoid such an issue; [48] we use this approach to initialize our method.

We define the upper bound $\tau := 5$ throughout our experiments, driven by discussions with medical experts and similarity to several medical scoring systems.

We define the fit of a solution given a certain target rank $R$ as the proportion of the total sum of squares that is explained by the$R$-dimensional model. [59] Formally, we have:

$$\text{Fit}\,(R) := 1 - \ \frac{\sum_{k=1}^{K} \left|\left|\mathbf{X_k} - \mathbf{U_k}\mathbf{S_k}\mathbf{V^T}\right|\right|_F^2}{\sum_{k=1}^{K} ||\mathbf{X_k}||_F^2}$$

We also define the phenotype sparsity measure as:

$$Sparsity(R) \ := \ 1 \ - \ \frac{\text{nnz}\,(\mathbf{V})}{\text{J} \quad \text{R}},$$

where $nnz(\mathbf{V})$ denotes the number of non-zero elements of the factor matrix $\mathbf{V}$, and $J \ R$ is the product of its mode sizes. Sparse solutions are considered more interpretable; thus, we prefer solutions where $Sparsity(R)$ is close to 1.

We finally define the phenotype overlap as the percentage of phenotype pairs sharing at least one common (overlapping) feature. For example, if $\mathbf{V}(i, \ k)$ and $\mathbf{V}(i, \ l)$ are both non-zero for $k \neq l$, then the pair of phenotypes $(k, l)$ shares $i$ as an overlapping feature. Lower values of overlap improve interpretability, as more distinct phenotypes can be extracted

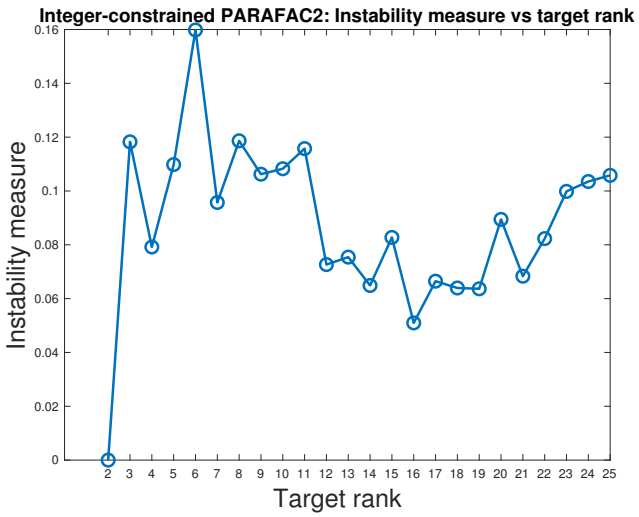



Figure 6.4: Guiding the choice of target rank for integer-constrained PARAFAC2. We repeat the experiments with 10 different initialization points for every target rank (R). The instability measure gives lower values to more robust solutions, i.e., where different initialization points for the same target rank give highly-correlated phenotype factors ($\mathbf{V}$), up to column permutations. Domain knowledge suggests that there are more than 2 distinct phenotypes for pre-HF; as a result, we choose the solution with $R = 16$.

which describe a wider range of clinical manifestations. Formally, we have:

$$\text{Overlap} := \frac{nnz(triu\left(\mathbf{V^T V}\right))}{\text{R } \frac{R-1}{2}}$$

We denote the upper-triangular (lower triangular is equivalent) part of the cross-product $\mathbf{V^T V}$ (excluding the main diagonal) as $triu(\mathbf{V^T V})$. We then count the number of non-zero elements and normalize by the maximum number of phenotype pairs.

**Choosing the number of phenotypes** We provide the results guiding our choice of the number of phenotypes in Figure 6.4. For each target rank $R \in \{2, \ldots, 25\}$, we run our proposed algorithm for 10 different initialization points. The instability measure gives lower values to more robust solutions, for which different initial points for the same target rank $R$ give highly-correlated phenotype factors$\mathbf{V}$, up to column permutations. Domain knowledge suggests that there are more than 2 distinct phenotypes describing the heterogeneity of pre-HF. Thus, we pick the solution with the second lowest instability measure, achieved for $R = 16$.

Table 6.2: 5 most prevalent phenotype definitions extracted via integer-constrained PARAFAC2

| Phenotype ID: 1 (Prevalence: 0.72) | Feature score |
|---|---|
| Dx_Essential hypertension [98.] | 5 |
| Med_ACE Inhibitors | 1 |
| Med_Beta Blockers Cardio-Selective | 1 |
| Med_Calcium Channel Blockers | 1 |
| **Phenotype ID: 13 (Prevalence: 0.59)** | **Feature score** |
| Dx_Immunizations and screening for infectious disease [10.] | 5 |
| Proc_Prophylactic vaccinations and inoculations | 4 |
| Dx_Medical examination/evaluation [256.] | 1 |
| Dx_Other screening for suspected conditions (not mental disorders or infectious disease) [258.] | 1 |
| Dx_Other upper respiratory disease [134.] | 1 |
| **Phenotype ID: 4 (Prevalence: 0.56)** | **Feature score** |
| Dx_Disorders of lipid metabolism [53.] | 5 |
| Med_HMG CoA Reductase Inhibitors | 2 |
| **Phenotype ID: 7 (Prevalence: 0.47)** | **Feature score** |
| Dx_Other skin disorders [200.] | 5 |
| Proc_Excision of skin lesion | 3 |
| Dx_Allergic reactions [253.] | 1 |
| Dx_Other and unspecified benign neoplasm [47.] | 1 |
| Dx_Other mycoses | 1 |
| Dx_Other non-epithelial cancer of skin [23.] | 1 |
| Proc_Other non-OR therapeutic procedures on skin and breast | 1 |
| **Phenotype ID: 14 (Prevalence: 0.45)** | **Feature score** |
| Dx_Other and unspecified lower respiratory disease | 5 |
| Dx_Chronic airway obstruction; not otherwise specified | 1 |
| Dx_Other and unspecified asthma | 1 |
| Med_Azithromycin | 1 |
| Med_Cough/Cold/Allergy Combinations | 1 |
| Med_Sympathomimetics | 1 |
| Proc_Routine chest X-ray | 1 |

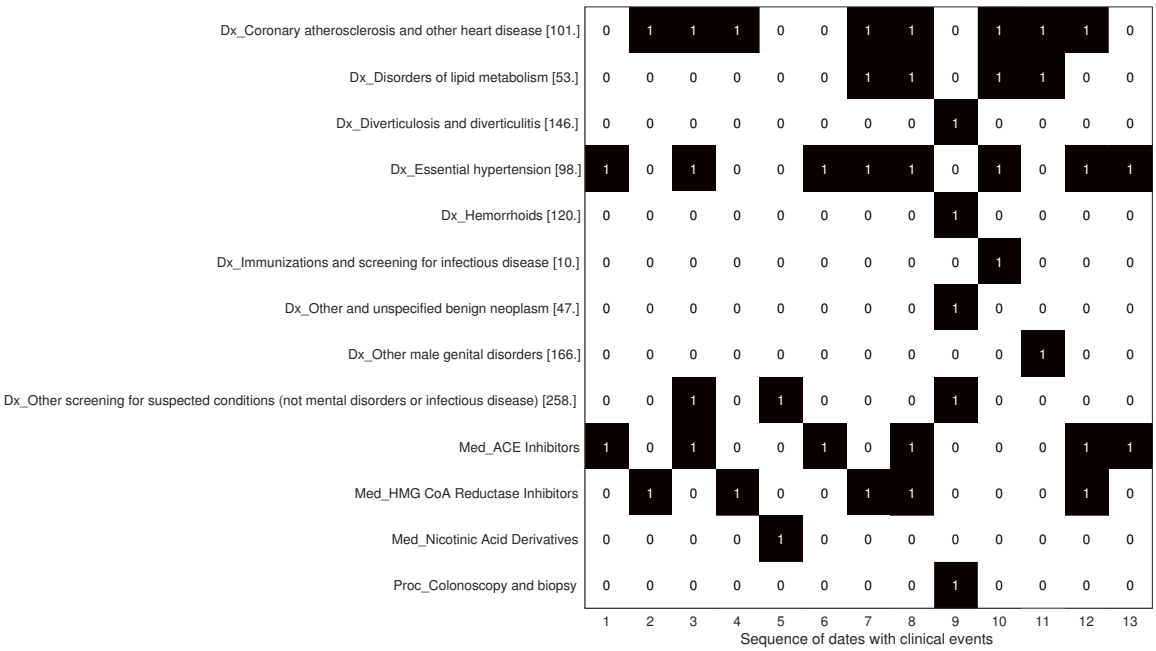



Figure 6.5: Example of input data for a single patient. The x-axis corresponds to the sequence of distinct dates of events and the y-axis corresponds to the diagnostic, medication and procedure categories. A value of 1 for the corresponding (clinical category, date) pair denotes that a clinical category was observed during that date in the patients records.

**Phenotype discovery and clinical evaluation** Table 6.2 provides the detailed definitions of the 5 most prevalent pre-HF phenotype extracted. For each phenotype, we have computed its prevalence (ranging within [0, 1]) across the HF cohort we examine. Prevalence is computed by inspecting the integer phenotype membership factor $\mathbf{S_k}$. We consider that the $k$-th patient exhibits the $j$-th phenotype if $\mathbf{S_k}\,(j, j) > 0$.

In Figure 6.6, we illustrate a heatmap of a scaled temporal signature for an example patient. Rows correspond to phenotype annotations and columns to distinct dates of medical records. We only list phenotypes with some non-zero value across any column; only 4 phenotypes are relevant for this patient. We picked a patient with a small number of distinct dates of events in order to facilitate validation by contrasting the result with the corresponding patient's input data (Figure 6.5). An example indicating the validity of the result is that the third row – corresponding to the phenotype containing "Coronary atherosclerosis and other heart disease [101.]" as a single feature – contains non-zero elements precisely during

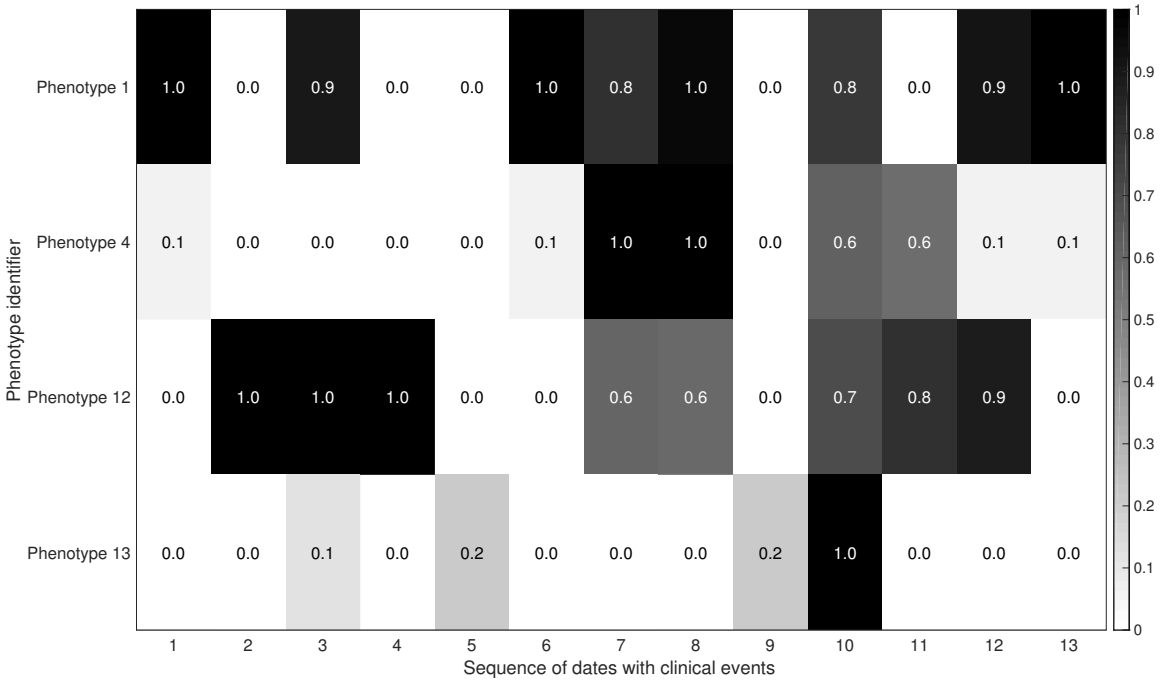



Figure 6.6: Illustration of a scaled temporal signature (row-normalized by the maximum row value) extracted via our proposed approach for the patient with input data shown in Figure 6.5. Rows correspond to phenotype definitions provided by a cardiologist and columns correspond to dates of clinical event occurrence. Phenotypes with some non-zero element in any date are only presented (i.e., only 4 phenotypes in total).

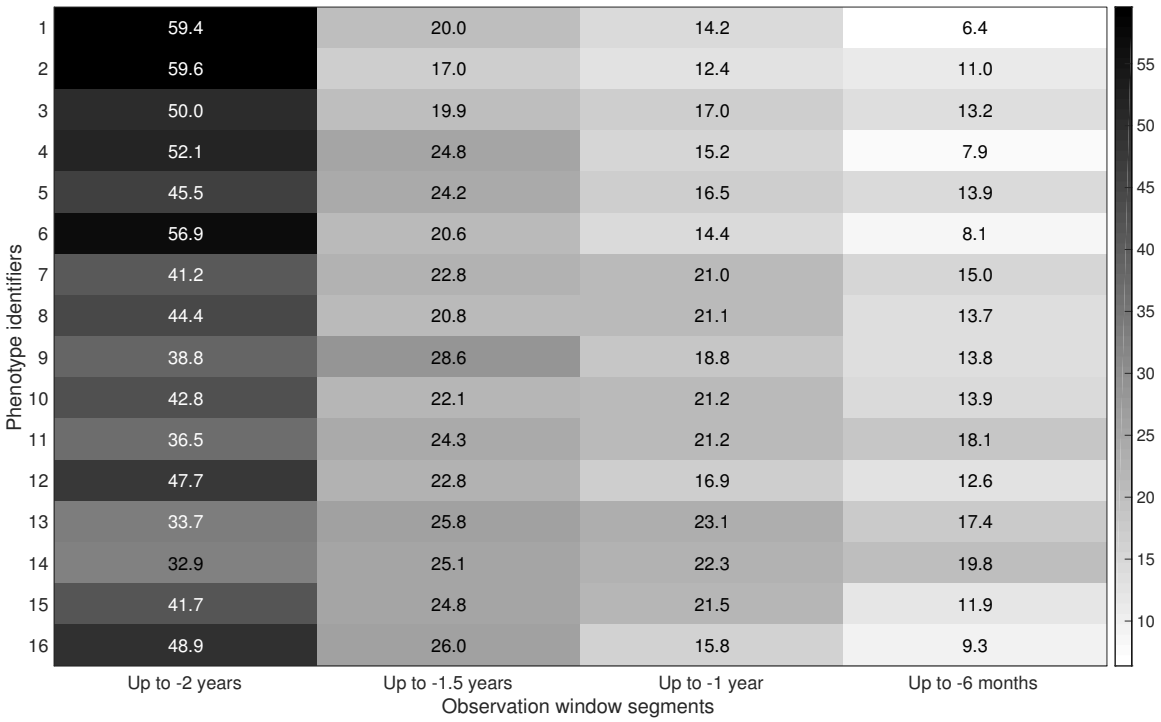



Figure 6.7: Percentages of earliest phenotype occurrences for every phenotype extracted. Each (i,j) cell corresponds to the percentage of patients developing the earliest occurrence of i-th phenotype during the j-th time segment (out of patients eventually developing the i-th phenotype during the observation window). For example, we remark that out of the 1864 patients developing Hypertension (phenotype identifier: 1), 1745 (94%) patients have developed it at least 1 year before the HF diagnosis date (i.e., before the last-most time segment we consider).

the dates of the corresponding diagnosis occurrence in the patient's records (Figure 6.5).

**Identifying earliest phenotype occurrence** In Figure 6.7, we provide with the percentages (each row sums to 100%) of earliest occurrence across all phenotypes and time segments. For example, the cell (1, 1) indicates that 59.4% of hypertensive patients have developed the Hypertension phenotype as early as 2 years before the HF diagnosis date.

There are potentially interesting insights revealed through this analysis which indicate differences in phenotype progression and occurrence in the EHRs. For example, out of the 1,864 patients developing Hypertension (phenotype index 1), 1,745 (94%) patients have developed it at least 1 year before the HF diagnosis date (i.e., before the last-most time segment we consider). On the other hand, out of the 1,149 patients developing lower res-

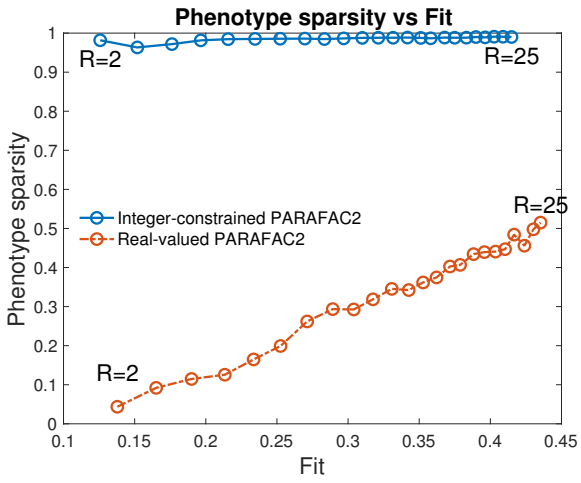



Figure 6.8: Comparing the proposed integer-constrained PARAFAC2 against its real-valued counterpart in terms of the phenotype sparsity fit trade-off. We run the experiments 10 times for each value of $R = 2, 3, , 25$ and choose the solution achieving the best fit. Sparsity reveals the amount of zero elements w.r.t. the total number of elements of the models matrix V, indicating the phenotype definitions. Sparsity is key for ease of interpretation; we remark that our proposed integer-constrained PARAFAC2 method achieves a much sparser solution without incurring a significant loss in terms of model fit.

piratory system disorders (Phenotype ID: 14), 227 of them (19.8%) exhibit this phenotype for the first time during the last-most time segment we consider.

We developed a simple validation method to test whether the earliest phenotype occurrence we extract as described in the "Materials and Methods" Section is not modeling noise, instead of a true phenotype presence. If a detection of the $j$-th phenotype occurs during the $i$-th sequenced date of the $k$-th patient's records, then we assess whether any of the features composing the $j$-th phenotype does occur during the $i$-th date of the k-th patient's input data. Based on this validation method, we measure precision (i.e., how many detections are relevant). Out of a total of 15,743 detections, 15,725 are relevant (precision: 99.9%).

**Quantitative comparisons in terms of model fit and interpretability** We quantitatively compare the proposed integer-constrained PARAFAC2 approach against its real-valued counterpart. [58, 61] We do so in terms of model fit and interpretability of the phenotypes' definition factor $\mathbf{V}$. Despite the fact that quantitatively measuring the interpretability of a

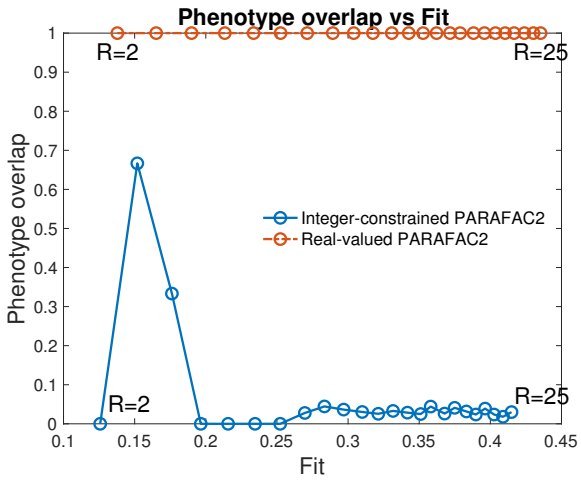



Figure 6.9: Comparing the proposed integer-constrained PARAFAC2 against its real-valued counterpart in terms of the phenotype overlap fit trade-off. We run the experiments 10 times for each value of $R = 2, 3, , 25$ and choose the solution achieving the best fit. Overlap is defined as the percentage of phenotype pairs having at least one overlapping feature (e.g., if $V(i,k)$ and $V(i,l)$ are both non-zero for $k \neq l$, then the pair of phenotypes $(k,l)$ shares $i$ as an overlapping feature). It is crucial that phenotype overlap is low for ease of interpretation; we remark that our proposed integer-constrained PARAFAC2 method achieves a much smaller value of feature overlap, without incurring a significant loss in terms of model fit.

machine learning model remains an open challenge, there are standard measures which are widely accepted as proxies for interpretability: a) model conciseness, which can be measured by assessing the model's sparsity, and b) minimization of inter-component overlap, in the sense that the different components captured (i.e., different columns of $\mathbf{V}$) should describe a wide range of latent concepts. The latter one is particularly relevant for phenotyping where different components should capture a wide range of clinical manifestations. All measures of fit, sparsity and overlap are described in the "Study design" Section.

As shown in Figure 6.8, our integer-constrained approach implicitly (i.e., it does not directly optimize for sparsity) achieves a much sparser solution than its real-valued counterpart across all different values of target rank $R \in \{2, \ldots, 25\}$. It does so without incurring a significant loss in terms of model fit (x-axis). In Figure 6.9, we also measure the phenotypes' definition $\mathbf{V}$ factor overlap. We remark that the real-valued PARAFAC2 model is noisy in the sense that all phenotypes overlap with each other (i.e., overlap value of 1 across

all values of target rank). We conclude that, in the absence of any arbitrary post-processing (e.g., truncation of features with the lower-most $k$ weights), the real-valued model does not lead to easily-interpretable solutions. On the contrary, our integer-constrained model is interpretable, without incurring a significant loss in terms of model fit.

## 6.5 Conclusion & Future Work

We proposed an integer-constrained PARAFAC2 tensor factorization to automatically identify pre-diagnostic phenotypes of heart failure patients through readily-available EHR data, as well as extract information about their temporal evolution. We quantitatively demonstrate the increased interpretability of the resulting phenotypes based on integer constraints, as compared to the corresponding real-valued model; to do so, we evaluate both in terms of model conciseness and feature overlap across different phenotypes. Potential future work may be to constrain factors $\{\boldsymbol{U}_k\}$ with non-negative/integer constraints in order to further improve interpretability. Utilizing the extracted low-rank integer patient representations as features in a predictive model tackling the challenge of early HF prediction in an interpretable manner is a direct next step of this work.

# CHAPTER 7

# USING THE PARAFAC2 TENSOR FACTORIZATION ON EHR AUDIT DATA TO UNDERSTAND PHYSICIAN DESKTOP WORK


Summary

In this Chapter, we explore how the EHR activity log file can be processed to derive scalable and quantifiable physician-level workflow measures. To cope with the massive volume of log data, we apply PARAFAC2 (as implemented via the SPARTan approach presented in Chapter 4) to identify clusters of audit log records that map to meaningful representations of physician tasks. The discovered variants of notes' access map to two known workflows for this task.


**Background:** Activity or audit log data are required for EHR privacy and security management but may also be useful for understanding desktop workflow**.**

**Objective:** We determined if the EHR audit log file, a rich source of complex time-stamped data on desktop activities, could be processed to derive primary care provider (PCP) level workflow measures.

**Methods:** We analyzed audit log data on 876 PCPs across 17,455 ambulatory care encounters that generated 578,394 time-stamped records. Each individual record represents a user interaction (e.g., point and click) that reflects all or part of a specific activity (e.g., order entry access). No dictionary exists to define how to combine clusters of sequential audit log records to represent identifiable PCP tasks. We determined if PARAFAC2 tensor factorization could: 1) learn to identify audit log record clusters that specifically represent defined PCP tasks; and 2) identify variation in how tasks are completed without the need for ground-truth labels. To interpret the result, we used the following PARAFAC2 factors: a matrix representing the task definitions and a matrix containing the frequency measure of

each task for each encounter.

**Results:** PARAFAC2 automatically identified 4 clusters of audit log records that represent 4 common clinical encounter tasks: 1) medications' access, 2) notes' access, 3) order entry access, and 4) diagnosis modification. PARAFAC2 also identified the most common variants in how PCPs accomplish these tasks. It discovered variation in how the notes' access task was done, including identification of 9 distinct variants of notes access that explained 77% of the input data variation for notes. The discovered variants mapped to two known workflows for notes' access and to two distinct PCP user groups who accessed notes by either using the Visit Navigator or the Wrap-Up option.

**Conclusions:** Our results demonstrate that EHR audit log data can be rapidly processed to create higher-level constructed features that represent time-stamped PCP tasks.

## 7.1 Introduction

Health care structure (e.g., facilities, staff, financing) defines the context for care processes. In ambulatory care, in particular, an increasing share of provider and staff time are spent at the desktop. [219] Log file data (also known as audit logs) may be used to reveal how desktop work is done and the time required to do this work. [220, 221, 222] Log files, however, are voluminous and difficult to decipher. We determined whether tensor factorization methods could be used to cluster log file records into groups that represent distinct PCP tasks. If this were possible, then these records could be used for a diversity of applications that include but are not limited to workflow analysis, efficiency improvement efforts, learning best practices for use of the EHR, and to better understand how work is done and improved.

EHRs automatically generate time-stamped audit logs that track all interactions that a user has with a patient's record, as well as, other desktop activities (e.g., inbox). While these data are created for privacy and security management they also document how users do their desktop work when using the EHR. [223] But, audit log data are inordinately com-

plex and are not organized into self-identifiable clusters that represent defined jobs or tasks. In this context, we use the term "job" as defined by Ulwick [224] to refer to the underlying goal of a sequence of actions being taken or tasks being completed. [223] For example, making a medication order in the EHR may be a task associated with the overall "job" of managing a patient's medication use for a specific condition with the intention of optimizing outcomes. The log file documents the electronic action (making the order), whereas the job is inferred from an overall ensemble of related actions (e.g., open the patient's medication list, search the medication database for the appropriate medication, select the dose/quantity/pharmacy) that may be summed and expressed as a series of tasks. If tasks can be readily identified, log files could be used to understand how desktop work is done, including the time required to do specific desktop job (i.e. make diagnosis, medication management, procedure order, progress note).

We determined if machine-learning methods could be used to interpret audit log data and rapidly identify the most common types of clinical desktop tasks and to determine variation in how these tasks are done.

## 7.2 Methods

### Overview

In this study, we used Epic audit log records from face-to-face primary care encounters to answer the following questions about what PCPs do during encounters [58, 59, 51] 1) Can raw time-stamped audit log records be organized into clusters of records that represent intuitively meaningful representations of tasks that PCPs do? 2) Can audit log records be automatically organized and interpreted to determine variation in how a specific task is done? For the latter, we specifically focused on progress notes because this work is done for every encounter. Moreover, it is known that there are two dominant ways to use the Epic EHR built-in interface features for this purpose. That is, clinical observation has confirmed that are two distinct methods by which progress notes are accessed and documented. We

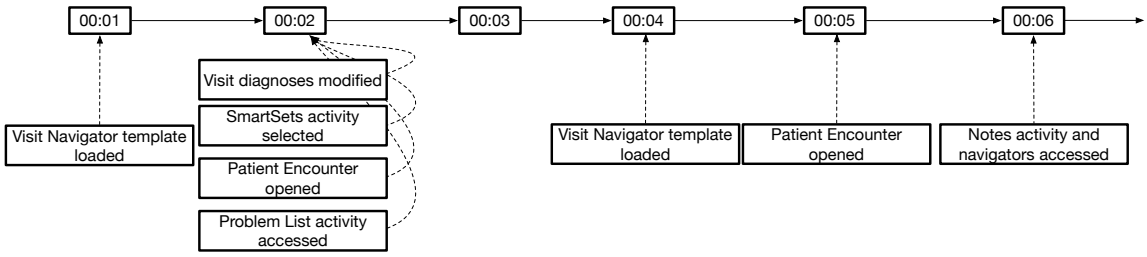



Figure 7.1: Example of EHR audit log records corresponding to the opening of the patient record at the start of an encounter. Note that there may be multiple log records with the same timestamp. Timestamps with no records are indicators of no computer related interactions (e.g., doctor is viewing data on screen, talking to patient, etc).

wanted to determine if machine learning could detect these two distinct methods and the ensemble of related tasks and also cluster PCPs by how progress notes were completed. To this end, the methods section describes the source data, pre-processing of the source data for use in modeling, and then the tensor factorization methods used to summarize audit log records.

**Source of Data & Data Extraction**

Audit log data were extracted from Sutter Health's Epic EHR data sources. Sutter Health is a large not-for-profit health system that serves over 100 communities in northern California. Sutter uses a single instance of the Epic Systems Corporation EHR.

Audit log files are very large even for a small number of encounters. We randomly selected two working days in 2016 (Epic 2015) and extracted log records for ambulatory care face-to-face encounters completed by 876 Primary Care Providers (PCPs). PCPs include physician, nurse practitioner, and physician assistants. The two days of log files resulted in 578,394 time-stamped audit log records across 17,455 encounters. Each entry in the audit log is time-stamped at a resolution of one-second, where two or more records occurring less than one second apart will have the same time-stamp, but these records are sequentially ordered as they occur. We restricted analysis to PCP face-to-face encounters that had at least two time-stamped audit log records. This resulted in 16,613 input encounters for PARAFAC2. We describe data pre-processing methods followed by the application of tensor factorization.

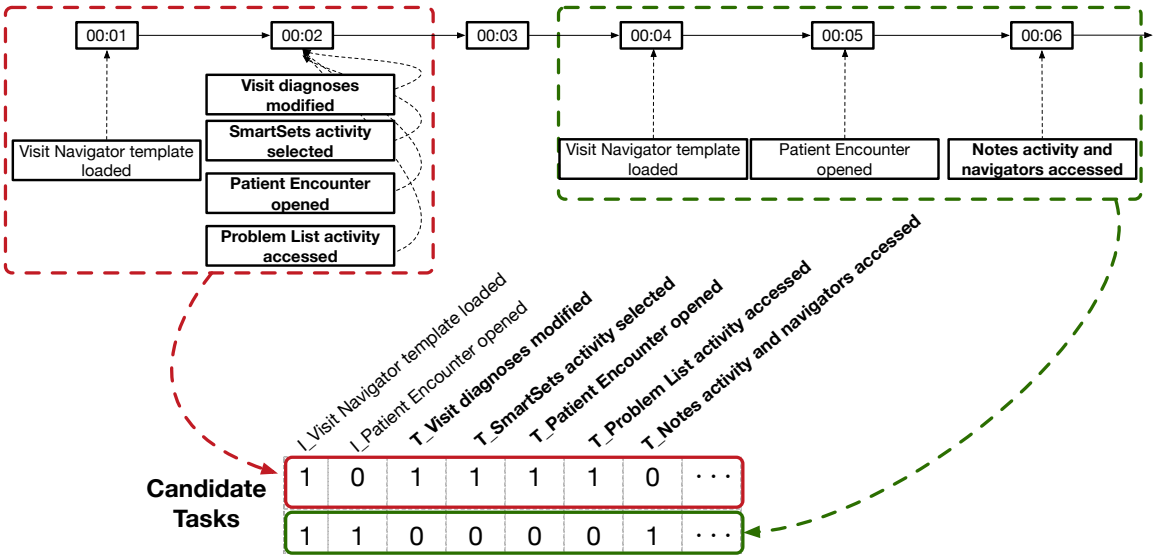



Figure 7.2: Transformation of face-to-face encounter audit log records data into a binary matrix. Each row of the matrix is a multi-hot vector encoding of the feature assembly representing a potential PCP task; the row number reflects the chronological task ordering during a single encounter. Each one of the columns corresponds to a feature defined as a variant of a certain audit log record: if an audit log record occurs or co-occurs with the last timestamp of a potential task, then this is marked as Terminal. Otherwise, it is marked as Intermediate. Terminal features are in bold font.

### Preprocessing of Raw Activity Log Data into Task and Feature Construction

We are interested in using tensor factorization to determine if a schema could be created where clusters of log records could be automatically identified that correspond to discrete tasks. To this end, log data were pre-processed by ordering them within provider, by date, by encounter within date, and last, by the audit log record time stamp and sequence indicator. Encounters that contained only one log record were removed. Figure 7.1 describes an audit log sequence that begins with opening of the encounter visit navigator, where multiple log records with the same timestamp means that they occurred during the same second.

Log records were further pre-processed to disassemble the contents of an encounter (sequence of log records) into smaller log record sequences that may reflect common tasks or parts of tasks done by PCPs. We refer to these discrete log record sequences as "potential tasks" that are defined as follows: in the absence of an explicit signal indicating the end of a task, we assume that sequential records that occurred on consecutive seconds (e.g., interval

00:04-00:06 in Figure 2) and that are separated by a break of no activity (i.e., $>1$ second) either before or after the sequence are considered to be part of the same potential task, but not necessarily representing a complete task.

Log records co-occurring with the last timestamp of a potential task indicate actions which occupy the user's attention (e.g., reviewing data retrieved when an EHR features is clicked), or that are consistently followed by a separate user activity (e.g., talking to the patient) that does not involve desktop interactions. Accordingly, we define two variants of audit log records which we call Intermediate (occurring before the last time stamp) and Terminal (co-occurring with the final timestamp before a $>1$ second gap between potential tasks) features. We annotate all records occurring during the $n$-th timestamp as Terminal features and records occurring during the timestamps $1, \ldots, n-1$ as Intermediate features. Also, duplicate features within the same potential task are removed. Such duplicates may arise when we create the multi-hot vector encodings of potential tasks as in Figure 7.2, when there may be multiple occurrences of a certain feature (e.g., I Visit Navigator template loaded).

In sum, we transform each $k$-th ambulatory care encounter to a binary matrix $\mathbf{X}_k$ of size $I_K \times J$, where $I_k$corresponds to the number of potential tasks for the $k$-th encounter and $J$ denotes the total number of features derived from raw audit log records (Figure 3). Importantly, the number of potential tasks per encounter is not known in advance. Finally, in the absence of labeled data, we do not know in advance whether a group of log records represents a systematic way of accomplishing a PCP task. PARAFAC2 tensor factorization [58] was used to address this challenge.

**PARAFAC2 Factorization: Variation in How PCPs Complete Tasks** We determined whether the PARAFAC2 factorization [58] could identify tasks and variation in how tasks are done. We selected the PARAFAC2 framework (i.e., model & factorization algorithm) because it can effectively account for the folded structure of our intermediate data. That is, our raw input data is from a set of PCP encounters, where the audit data from each rep-

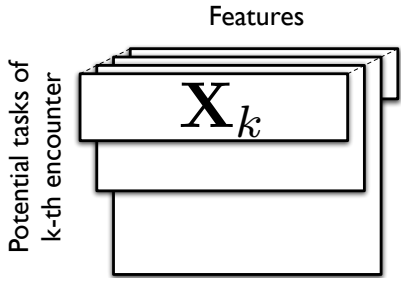



Figure 7.3: A visual representation of the collection of matrices created through our proposed method of organizing raw activity log records into candidate physician tasks. Each matrix corresponds to a single encounter. The rows correspond to candidate physician tasks recorded (which can vary across encounters) and the columns to the features constructed from raw activity log records.

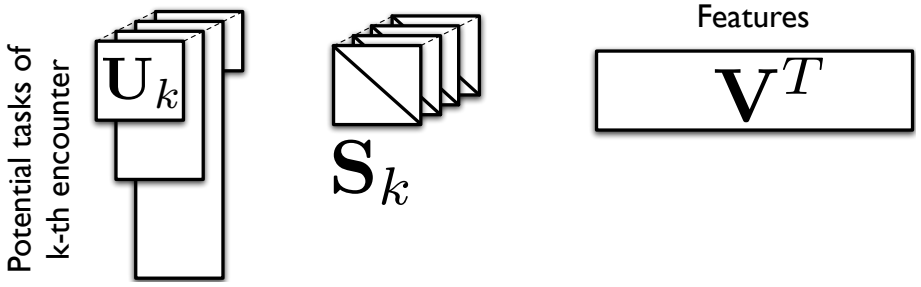



Figure 7.4: PARAFAC2 factorization given an input collection of matrices (as in Figure 7.3). The chosen model can naturally handle a varying number of candidate tasks across different encounters. The matrix $\boldsymbol{V}$ contains the most prevalent variation patterns of the most common physician tasks.

resents a sequence of records corresponding to underlying potential tasks, each of which can ultimately be represented by an ensemble of features (i.e., audit data entries). Such a folded structure can be concisely modeled by a multi-dimensional data array, containing a matrix of sequenced potential tasks X features for each one of the PCP encounters; such an array fits the expected input for the PARAFAC2 factorization. PARAFAC2 also offers an easily-interpretable approach to dimensionality reduction, as noted below. Notably, in contrast to other simpler multi-dimensional dimensionality reduction approaches (e.g., the CP/PARAFAC decomposition [89]), PARAFAC2 does not require any ad-hoc alignment of tasks among encounters. This is crucial because it is unrealistic to assume that all encounters will share the same number of potential tasks. PARAFAC2 also preserves model uniqueness under mild assumptions, boosting reliability of model interpretation (we elaborate on that below) and also it natively handles input data sparsity, which is a by-product of the fact that a few features will be observed for every potential task. Finally, specialized algorithms have recently been developed [51] scaling up PARAFAC2 for sparse input data; the use of scalable frameworks is crucial in order to cope with the voluminous nature of log files.

The input data for this decomposition (Figure 7.4) are organized as a binary multi-way array (tensor) of size $I_k \times J \times K$ where $K$ is the number of encounters. PARAFAC2 approximates a collection of matrices $\{\boldsymbol{X}_k \in \mathbb{R}^{I_k \times J}\}$ as:

$$\boldsymbol{X}_k \approx \boldsymbol{U}_k \, \boldsymbol{S}_k \, \boldsymbol{V}^T \tag{7.1}$$

where $k = 1, \ldots, K, \boldsymbol{U}_k \in \mathbb{R}^{I_k \times R}, \boldsymbol{S}_k \in \mathbb{R}^{R \times R}$ is diagonal and $\boldsymbol{V} \in \mathbb{R}^{J \times R}$. $R$ is a user-defined parameter which denotes the number of tasks extracted from the model. We exploit the following factors of the PARAFAC2 model: a) a matrix revealing the most representative variations of physician task definitions across all encounters ($\boldsymbol{V}$ matrix in the model), and b) a matrix containing the frequency of each task extracted for each k-th encounter

($\boldsymbol{S}_k$ matrix in the model). The model can be extended so that non-negative constraints are imposed on $\{\boldsymbol{S}_k\}, \boldsymbol{V}$ factors, [145] a constraint we employ to enhance interpretability.

**Model and Algorithm** PARAFAC2 preserves uniqueness of the model output by imposing the constraint that the cross product $\mathbf{U}_k^T\mathbf{U}_k$is invariant regardless of which k is involved. [58] This implies that the correlations between the tasks extracted are constant for all the encounters. [59] Preserving uniqueness means that the model is pursuing the *actual latent factors,* rather than an arbitrary rotation of them, which boosts the reliability of the model interpretation. [131]

The state-of-the-art framework for fitting the PARAFAC2 model uses Alternating Least Squares (ALS). [59] This corresponds to optimizing a least-squares criterion, in an alternating fashion, where we solve for a subset of factors by fixing all others. Our input data are inherently sparse, meaning that there are a few non-zero elements recorded as compared to the total size of the input matrices. Thus, we use the algorithm proposed by Perros et al., which also follows the ALS framework, but is optimized for sparse datasets. [51, 61]

To avoid repetition, we refer the reader to the Section 5.3 regarding the details of determining the number of task definitions via utilizing the Core Consistency Diagnostic [171, 190]. To increase our chances of avoiding local minima in our experiments, we ran the PARAFAC2 fitting algorithm 10 times for each target rank $R$ (i.e., number of task definitions pursued) using random initialization of factors and with the Singular Value Decomposition (SVD)-based initialization suggested in [154]. Then, we chose the solution with the minimum cost function error, among the ones achieving a diagnostic score of at least 90%. If the diagnostic score was lower than 90% for all 10 runs, then we simply picked the result with the best diagnostic score.

On the other hand, we define the percentage of fit as the proportion of the total sum of squares that is explained by the model: [59]

$$\text{Fit}\,(R) := \; 1 - \; \frac{\sum_{k=1}^{K} \left|\left|\mathbf{X}_k - \mathbf{U}_k\, \mathbf{S}_k\, \mathbf{V^T}\right|\right|_F^2}{\sum_{k=1}^{K} ||\mathbf{X}_k||_F^2}$$

A reasonable expectation is that as long as we increase the number of representative task definitions pursued, the fit percentage will increase as we are providing the model with more factors to describe data variation.

**Proposed Model Interpretation** Below, we summarize the proposed approach to interpret the model parameters. Each one of the $R$ columns of the $\boldsymbol{V}$ matrix of the model provides a certain task definition  to extract this definition, we order the weights of each column in decreasing order and list the associated features. Due to the non-negativity constraints imposed on $\boldsymbol{V}$, features associated with higher weights are expected to be more representative of the corresponding task definition.

The diagonal $\boldsymbol{S}_k$ matrix in the model is representative of the frequency of each task extracted for each k-th encounter. To estimate the task utilization for each user (i.e. clinician), we propose to consider the mean of all the $R$-dimensional encounters for each user. The resulting $R$-dimensional vector for each user can be used to identify variations among the users, with respect to task utilization. To visually identify clusters of users with similar utilization profiles, we use the tSNE [194] software, which can reduce the $R$-dimensional vectors to the 2-dimensional space.

### 7.3 Results

For our experiments, we used the Matlab implementation of the PARAFAC2 algorithm designed for sparse input, provided in [51]. Among a total of 84 features, derived as Intermediate or Terminal variants of audit log records (refer to the Methods Section for details), we excluded 36 that accounted for less than $1\%$ of all feature occurrences across all the encounters. This resulted in $J = 48$ unique features as input to the model. No significant change in fit (less than $1\%$) was observed before and after excluding the 36 uncommon features.

Figure 7.5 summarizes the percentages of fit and consistency diagnostic for various target ranks (i.e., number of tasks). As suggested in [190], the final solution comprising

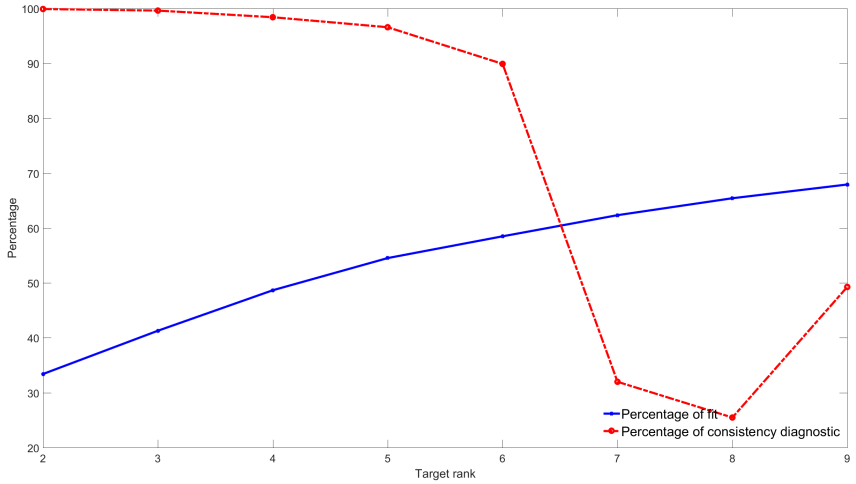



Figure 7.5: The percentage of consistency diagnostic and fit. We choose the solution providing the highest fit (i.e., lowest error) and also providing a well-specified model reflected by the consistency diagnostic (over 90 percent). Thus, the best solution for 5 task definitions is selected.

Table 7.1: Task definitions discovered via PARAFAC2. The first part of each entry denotes whether the feature corresponds to an activity log record that is an intermediate (I) or a terminal one (T). The terminal activities are in bold font. Activities are listed in order of frequency of occurrence as indicated by the PARAFAC2 model weights. Activities associated with weights $> 0.2$ are listed.

| Task | Features |
|---|---|
| 1. Medication Access | **T_Medications activity accessed**<br>**T_Patient Encounter opened**<br>I_Visit Navigator template loaded<br>**T_Chief Complaint navigator section accessed**<br>**T_PCP History accessed**<br>I_Data from related encounters accessed through VB |
| 2. Note Access | **T_Notes activity and navigators accessed**<br>**T_Patient Encounter opened**<br>**T_Problem List activity accessed** |
| 3. Order Access | **T_Order Entry activity accessed**<br>**T_Diagnosis association updated** |
| 4. Diagnosis Modification | **T_Visit diagnoses modified**<br>**T_Patient Encounter opened**<br>I_Visit Navigator template loaded |

five tasks has the highest fit, as well as, a high consistency diagnostic (above 90%). Thus, for our experiment, the solution with 5 tasks is considered the most suitable.

Table 7.1 presents the most common tasks discovered using PARAFAC2 in our first experiment. The name assigned to each task was derived from the log record with highest weight in each cluster. We have used the term "access" in Table 7.1 as a general term for the cluster of EHR features accessed by the user; this does not imply that the task is a read-only view of the EHR; the audit log source file does not record more granular activities, such as keyboard typing, data entry, copy/paste activities, etc.

**Discovery of most representative PCP tasks** Without prior information, PARAFAC2 identified five groups of features accounting for 55% of the variation in activity log data (Figure 7.5). One of those 5 groups contained intermediate features as the ones having the largest frequency of occurrence and thus was excluded from our result. For each of the remaining 4 discovered task definitions, we list the identified features, in order of frequency as indicated by the PARAFAC2 model weights, and also indicate whether they are intermediate (I) or terminal (T). We annotate each one of the feature groups according to their most frequent terminal feature (e.g., we consider the first feature group in Table 7.1 to reveal the most common variation pattern of completing the Medication access task).

**Variation in Notes Access Task** We further determined whether PARAFAC2 could be used to extract variation in activity log records that correspond to Notes Access (Table 7.1) with the intention of detecting differences among physicians in how notes are accessed and done.

Beginning with the 17,455 unique input encounters, we only retained tasks that contained the T_Notes activity and navigators accessed feature, indicating the inclusion of Notes access as a terminal activity. We then follow the same pre-processing strategy as above, by excluding very infrequent features ($< 1\%$ of the total feature frequency, N=14,659) and excluding 51 encounters that contained a single activity.

Figure 7.6 summarizes the percentages of fit and consistency diagnostic for various

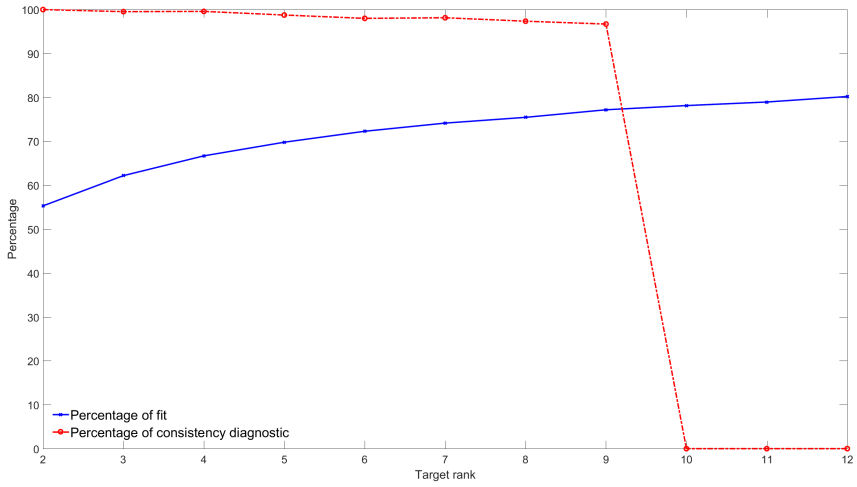



Figure 7.6: Plotting the percentage of consistency diagnostic and fit for the Notes activity access. We choose the solution providing the highest fit (i.e., lowest error) and also providing a well-specified model reflected by the consistency diagnostic (over 90 percent). Thus, the best solution for 9 notes access variation patterns is selected. Note that for illustration purposes, we have set the values of consistency diagnostic less than zero to zero.

target ranks (i.e., number of pursued notes access variation patterns). We extracted 9 distinct patterns associated with the notes access task that explained approximately 77% of the input data variation.

We examined whether the 9 notes access pattern variants were specific to physician groups using the pattern utilization indicator model factor ($\boldsymbol{S}_k$ in Figure 7.4). This factor describes the pattern utilization of each k-th encounter for all R variation patterns extracted. To estimate the pattern utilization for each user (i.e. clinician), we simply estimate the mean of all the R-dimensional encounters for each user. We used the tSNE [194] software to visualize user representations, by reducing the 9-dimensional vectors to the 2-dimensional space.

Figure 7.7 shows a clear separation of clinicians into two distinct groups (denoted as Groups A and B on the left and bottom right part of Figure 7.7 respectively). We separately examined the log records of group A and B to understand notes related activities and task patterns.

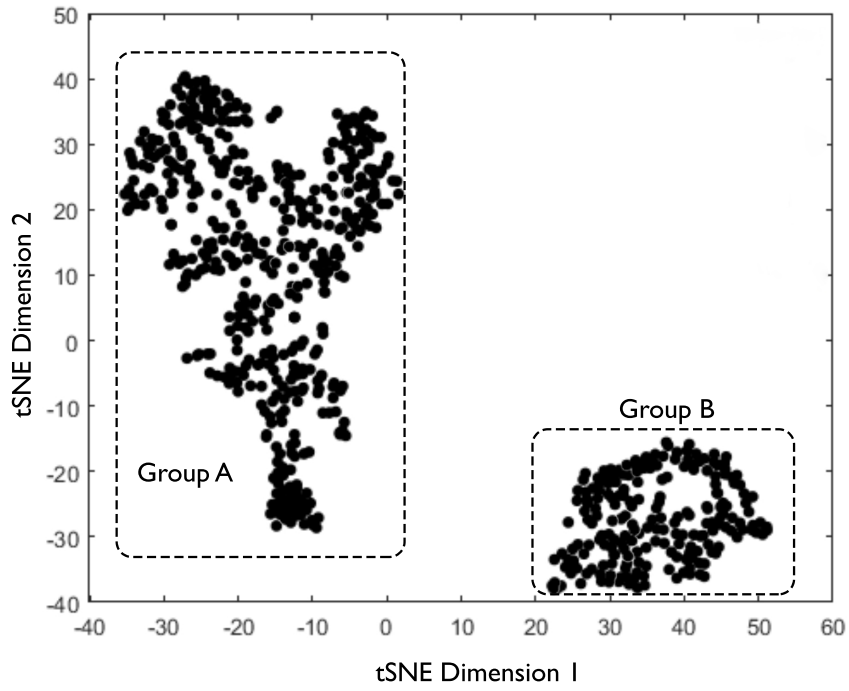



Figure 7.7: tSNE visualization of the PCP patterns for Notes access. Each point in the 2-D space corresponds to a PCPs relative position with respect to her pattern of utilization. The key conclusion is that there do exist 2 distinct clouds of points corresponding to 2 groups of primary care providers, based on the 9 variation patterns of notes access. Individual axes can be safely ignored, as tSNE only preserves neighborhood information between data points.

Groups A and B were distinct in that the 6th variation pattern is the least used by group A (4%), but is often used by group B (29%). The common activities for group B clinicians is SmartSets activitity selected, which corresponds to a Wrap-Up tag in the EHR that is available to directly access notes before closing a patient encounter in the EHR.

Group B clinicians rarely used the patterns 2, 5, and 7 (Figure 7.7). A common activity of those patterns is the loading of the Visit Navigator, which is used by 30% of group B clinicians, and only 0.5% of group A clinicians. The latter corresponds to use of the Visit Navigator tag to access progress note in the EHR.

Finally, we independently worked with an EPIC expert and asked for a demonstration of ways to access clinic notes. For each approach that was demonstrated we recorded the underlying log record sequence of EPIC interface tags. Two dominant approaches to access clinic notes were identified, one via the "Visit navigator" tag and one via "Wrap-Up" tag.

## 7.4 Discussion

Electronic health records (EHR) data are increasingly used to understand the relation between the care that was delivered and the change in patient outcomes. But, workflow improvement efforts have a different focus, attempting to increase efficiency and effectiveness of work processes within and between encounters and to monitor progress towards efficiency. [225] Clinical data, alone, are usually insufficient to understand whether workflow efforts like "Lean process improvement" is working and, importantly, to understand how it is working. In fact, a common criticism of healthcare workflow improvement work is that it lacks scientific measurement principles supported by readily accessible continual data. Audit log data may be useful in providing a scientific foundation for workflow management. Clinic practice workflow improvements (e.g., Lean principles) have not consistently translated into sustainable efficiency improvements. [226] In part, this may be because workflow efforts in health care lack a systematic and scalable approach to quantifying processes before and after a workflow improvement are implemented. [219, 227]

How work gets done has been described by Ulwick [224] and Christensen [228] as fundamentally important to understanding how to innovate in ways that have a high probability of success. [228, 229] While these methods have been used in a diversity of business sectors, little work has been done in healthcare. The volume and availability of audit log files offers a means to understand how healthcare is delivered by PCPs and others and to understand how work actually gets done. Access to this high-resolution data may offer a means to accelerate innovations to making healthcare work easier and more effective.

Audit log file data have not been widely used to understand care processes. Adler-Milstein et al. examined the impact of EHR platform usage on PCP productivity. [230] Tai-Seale et al. analyzed the time allocation patterns of PCPs between seeing patients and using the EHR platform. [231] Hirsch et al. used audit log file data to better understand clinical workflow practices and the corresponding patient experience. [223] Hribar et al. exploited the audit log file to calculate timings of clinical workflows. [232] Wetterneck et al. present a PCP task list derived from human observations of encounters; such an approach though has limited scalability due to the laborious nature of manual encounter observations. [233] None of these previous studies used audit log data to reveal PCP tasks and to understand how tasks get done (e.g., workflow variation patterns), nor to enable classification of EHR users based on their EHR access variation patterns to complete clinical tasks. Our work is a first step towards exploiting audit logs to understand PCP work processes. To this end, we applied advanced machine learning methods to identify the most common tasks that PCPs do during office encounter, as well as, common variants of those tasks. Our work opens the possibility of developing scalable quantitative measurement tools to both understand clinical tasks and the time required for these tasks.

One logical next step to this work is to link the task variation patterns with PCPs to clinical efficiency measures (e.g. minimal time spending in specific EHR area): given identification of user EHR utilization patterns to complete tasks, we could then estimate which group is more efficient in term of time spent in the tasks and timely completeness

of encounter documentation. This type of work could offer rapid means of improving how EHRs could be more effectively used by PCPs, as well as, identifying EHR tasks that are a challenge to complete or too time-consuming, as priority candidates for vendor improvements.

Audit log data metrics could also be linked with clinical outcomes for defined patient segments that vary by disease burden to study variation in clinical processes and demands. In so doing, it may be possible to identify optimal approaches to use of the EHR for patient subgroups. For example, patients with severe or complicated conditions require much more time and effort by the PCPs during the encounter than less severe patients. Examination of audit logs for encounters of patients who are very demanding may reveal specialized workflows that facilitate the PCP's use of the EHR and an increase in the volume of work done with less effort.

Understanding audit log data is in its infancy. Our work and that of others is only beginning to reveal the potential power of these data. Our preliminary use of PARAFAC2 revealed only five tasks. These were the most common. Future work will likely involve an iterative process where the most common tasks are identified and documented. Once these tasks are identified, the audit log records for these known, common tasks can be eliminated so that less common tasks can be revealed. To this end, we recently eliminated the audit log records associated with these first five tasks and then applied PARAFAC2 to the remaining log records. Not surprising, a new set of tasks were identified. While we partially demonstrated this methodology by focusing on the Notes' access variation patterns, a systematic application of this idea is an important future direction to pursue in ultimately creating an ontology that could be used to represent the content of audit log data.

## 7.5 Limitations

There are several limitations to the current work. We did not consider the frequency of features within a task in each of our experiments. Duplicate features may represent ˜5% of

consecutive activities. Retaining repeated features may increase the variation of patterns, but it does not impact the identification of features that were used to represent a pattern variation. The current task variation patterns captured ˜55% of EHR audit log record variation; We have not investigated the remaining 45% of variation, other than to note that iterative application of PARAFAC2 reveals new tasks. As we have noted, removal of the audit log records associated with these common tasks with re-application of PARAFAC2 to the remaining log records identifies less common tasks. In addition to the four well-defined clinical tasks we have identified, in-basket work (i.e. managing patient phone call, order results, order authorization, emails with patients or other PCPs, referral, etc.) accounts for 10-15% of PCP time. But, in-basket work is often done in fragments where individual tasks are interspersed among others. This makes it difficult to dissect these tasks from log records. Future work on iterative extraction of tasks may reveal the details of how in-basket work is done as a step towards reducing the growing demand that this work imposes on PCPs.

In the absence of a signal or dictionary indicating the end of a task, we adopted the simplifying assumption that all log records that are separated by less than a second are part of the same potential task. While we acknowledge this simplification, we argue that our approach fully utilizes log records at the available time-stamped resolution (i.e. one-second) and is robust to the two scenarios that may occur from this assumption. First, records grouped together may not belong to the same potential task. For example, in Figure 7.2, it is possible that "Visit Navigator template loaded" is an irrelevant log record to the second potential task (timestamps 00:04-00:06). Still, PARAFAC2 would still capture the (I_Patient Encounter Opened, T_Notes activity and navigators accessed) task if this was prevalent in other encounters, whether or not it had the same timestamp as other features in the set. Second, groups of records that should be collectively considered to indicate a single task may be assigned to two different tasks by the model. For example, in Figure 7.2, it is possible that both of the potential tasks we extracted should actually be grouped together

to form a single task. Still, PARAFAC2 would capture more granular tasks if log records composed to the tasks are commonly clustered together in other encounters.

Finally, clinical validation is needed to ensure that the labels we have applied to each identified task indeed map to what a PCP is trying to accomplish following the corresponding log record sequence. However, we believe that assigning the label based on the log record with the highest weight (thus, indicating the most frequent record), as we did, is the most reasonable data-driven way to label each task.

## 7.6 Conclusions

Our application of PARAFAC2 to audit log data offers an analytic approach that can facilitate the creation of ontologies or libraries that describe how healthcare work is done. We believe this is a critical step towards creating innovations that successfully equip PCPs and others to get more done with less effort and with a better experience.

# CHAPTER 8
# CONCLUSION AND FUTURE WORK

In this dissertation, we demonstrate how tensor factorization can be applied and extended to tackle several important problems in healthcare (unsupervised EHR-based phenotyping and automating understanding of physician desktop work) and biomedicine (drug-perturbed, tissue-specific gene expression prediction). In the following of this Chapter, we specify future work directions focusing on EHR-based phenotyping, which is a central topic of this dissertation.

**Downstream healthcare tasks.** Most of unsupervised phenotyping research (including the Chapters 3, 4, 5, 6 presented as part of this dissertation) has focused on extracting clinically-meaningful phenotype definitions. Utilizing the computed patient representations and phenotype definitions for downstream healthcare tasks is a ripe target for future work. The low-rank patient representations computed can be used as features for predictive modeling (e.g., towards early detection of heart failure which can provide with actionable insights regarding treatment development and management [234]). A predictive model based on the low-rank patient representations would provide a more clinically-meaningful, thus interpretable, result as it would utilize a small set of higher-level validated features indicating concrete patient phenotypes, as opposed to a large set of raw and noisy input features. Previous work [24] has demonstrated proof-of-concept success towards interpretable predictive modeling by collapsing the patient history to aggregate indicators of medical code records. A next step is to assess whether modeling the longitudinal evolution of patient records (as in Chapters 4, 5, 6) could provide patient representations with higher predictive power. Utilizing the results of unsupervised phenotyping to identify similar patients and inform clinicians in the absence of gold-standard evidence is an additional target for future work. Most of recent research towards this direction has focused on rule-based

phenotypes [6], which are limited by the substantial expert supervision required to create and maintain them.

**Robust unsupervised phenotyping.** Patients with the same named disease may differ in the number or density of medical events, due to differences in disease severity, ability to self-manage, interest in self-managing, poor access to care and other reasons. In essence, there may be outlier patients (e.g., with many more encounter records than others) which could obfuscate the patient phenotypes discovered. Prior work [235] tackles the challenge of robust tensor factorization for the CP tensor model. Developing robust unsupervised phenotyping methods which can handle the longitudinal evolution of patient data (e.g., via the PARAFAC2 model) is a target for future work.

**Incorporating additional EHR data domains.** Finally, our work has been mostly focusing on utilizing structured code information from the EHR. Including clinical text as well as continuous values such as lab test information is an additional target for future work. Modeling continuous lab values is particularly challenging for tensor-based approaches, as this would render the result harder to interpret than with categorical input data.

## REFERENCES


[1] N. R. Clark, K. S. Hu, *et al.*, "The characteristic direction: A geometrical approach to identify differentially expressed genes," *BMC bioinformatics*, vol. 15, no. 1, p. 1, 2014 (cit. on pp. 1, 4).

[2] M. Sirota, J. T. Dudley, *et al.*, "Discovery and preclinical validation of drug indications using compendia of public gene expression data," *Science Translational Medicine*, vol. 3, no. 96, 96ra77–96ra77, 2011 (cit. on pp. 1, 4, 14).

[3] W.-Q. Wei, P. L. Teixeira, H. Mo, R. M. Cronin, J. L. Warner, and J. C. Denny, "Combining billing codes, clinical notes, and medications from electronic health records provides superior phenotyping performance," *Journal of the American Medical Informatics Association*, vol. 23, no. e1, e20–e27, 2015 (cit. on pp. 1, 5).

[4] C. A. Longhurst, R. A. Harrington, and N. H. Shah, "A green buttonfor using aggregate patient data at the point of care," *Health affairs*, vol. 33, no. 7, pp. 1229–1235, 2014 (cit. on pp. 1, 5, 107).

[5] S. R. Thadani, C. Weng, J. T. Bigger, J. F. Ennever, and D. Wajngurt, "Electronic screening improves efficiency in clinical trial recruitment," *Journal of the American Medical Informatics Association*, vol. 16, no. 6, pp. 869–873, 2009 (cit. on pp. 1, 5, 9, 90).

[6] S. Gombar, A. Callahan, R. Califf, R. Harrington, and N. H. Shah, "It is time to learn from patients like mine," *Npj Digital Medicine*, vol. 2, no. 1, p. 16, 2019 (cit. on pp. 1, 107, 152).

[7] J. M. Banda, M. Seneviratne, T. Hernandez-Boussard, and N. H. Shah, "Advances in electronic phenotyping: From rule-based definitions to machine learning models," *Annual Review of Biomedical Data Science*, vol. 1, no. 1, pp. 53–68, 2018. eprint: `https://doi.org/10.1146/annurev-biodatasci-080917-013315` (cit. on pp. 1, 6, 7, 89, 92, 107, 108).

[8] X. Wang, D. Sontag, and F. Wang, "Unsupervised learning of disease progression models," ser. KDD '14, New York, NY, USA: ACM, 2014, 8594 (cit. on pp. 1, 8, 92, 115).

[9] H. Paik, M. J. Kan, N. Rappoport, D. Hadley, M. Sirota, B. Chen, U. Manber, S. B. Cho, and A. J. Butte, "Tracing diagnosis trajectories over millions of inpatients reveal an unexpected association between schizophrenia and rhabdomyolysis," *BioRxiv*, p. 473 082, 2018 (cit. on pp. 1, 8).

[10] A. Dagliati, L. Sacchi, A. Zambelli, V. Tibollo, L Pavesi, J. H. Holmes, and R. Bellazzi, "Temporal electronic phenotyping by mining careflows of breast cancer patients," *Journal of biomedical informatics*, vol. 66, pp. 136–147, 2017 (cit. on pp. 2, 8, 90, 94, 112, 115).

[11] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009 (cit. on pp. 2, 15, 45, 62, 63, 65, 66, 68, 111).

[12] J. Kim, Y. He, and H. Park, "Algorithms for nonnegative matrix and tensor factorizations: A unified view based on block coordinate descent framework," *Journal of Global Optimization*, vol. 58, no. 2, pp. 285–319, Feb. 2014 (cit. on pp. 2, 41, 49, 52).

[13] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, 2017 (cit. on p. 2).

[14] E. E. Papalexakis, C. Faloutsos, and N. D. Sidiropoulos, "Tensors for data mining and data fusion: Models, applications, and scalable algorithms," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8, no. 2, p. 16, 2017 (cit. on pp. 2, 120).

[15] T. G. Kolda and J. Sun, "Scalable tensor decompositions for multi-aspect data mining," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, IEEE, 2008, pp. 363–372 (cit. on pp. 2, 34).

[16] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix tri-factorizations for clustering," in *KDD 2006*, pp. 126–135 (cit. on pp. 2, 20, 34).

[17] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, Oct. 1999 (cit. on pp. 2, 34, 35, 40).

[18] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, "Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering," in *Proceedings of the fourth ACM conference on Recommender systems*, ACM, 2010, pp. 79–86 (cit. on pp. 2, 34).

[19] E. Acar, C. Aykut-Bingol, H. Bingol, R. Bro, and B. Yener, "Multiway analysis of epilepsy tensors," *Bioinformatics*, vol. 23, no. 13, pp. i10–i18, 2007 (cit. on pp. 2, 34).

[20] A. H. Williams, T. H. Kim, F. Wang, S. Vyas, S. I. Ryu, K. V. Shenoy, M. Schnitzer, T. G. Kolda, and S. Ganguli, "Unsupervised discovery of demixed, low-dimensional neural dynamics across multiple timescales through tensor component analysis," *Neuron*, 2018 (cit. on pp. 2, 120).

[21] J. C. Ho, J. Ghosh, and J. Sun, "Marble: High-throughput phenotyping from electronic health records via sparse nonnegative tensor factorization," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '14, New York, NY, USA: ACM, 2014, pp. 115–124 (cit. on pp. 2, 7, 34, 35, 58).

[22] Y. Luo, Y. Xin, E. Hochberg, R. Joshi, O. Uzuner, and P. Szolovits, "Subgraph augmented non-negative tensor factorization (santf) for modeling clinical narrative text," *Journal of the American Medical Informatics Association*, vol. 22, no. 5, pp. 1009–1019, 2015 (cit. on pp. 2, 5).

[23] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *Journal of Mathematics and Physics*, vol. 6, no. 1, pp. 164–189, 1927 (cit. on pp. 2, 44, 62, 65, 95).

[24] J. C. Ho, J. Ghosh, S. R. Steinhubl, W. F. Stewart, J. C. Denny, B. A. Malin, and J. Sun, "Limestone: High-throughput candidate phenotype generation via tensor factorization," *Journal of biomedical informatics*, vol. 52, pp. 199–211, 2014 (cit. on pp. 3, 7, 35, 58, 62, 90, 108, 111, 114, 151).

[25] C. O. S. Sorzano, J. Vargas, and A. P. Montano, "A survey of dimensionality reduction techniques," *ArXiv preprint arXiv:1403.2877*, 2014 (cit. on p. 3).

[26] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006 (cit. on p. 3).

[27] C. Olah, A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, and A. Mordvintsev, "The building blocks of interpretability," *Distill*, vol. 3, no. 3, e10, 2018 (cit. on p. 3).

[28] M. Bredel and E. Jacoby, "Chemogenomics: An emerging strategy for rapid target and drug discovery," *Nature Reviews Genetics*, vol. 5, no. 4, p. 262, 2004 (cit. on p. 3).

[29] J. A. DiMasi, R. W. Hansen, and H. G. Grabowski, "The price of innovation: New estimates of drug development costs," *Journal of health economics*, vol. 22, no. 2, pp. 151–185, 2003 (cit. on p. 4).

[30] J. T. Dudley, T. Deshpande, and A. J. Butte, "Exploiting drug–disease relationships for computational drug repositioning," *Briefings in bioinformatics*, vol. 12, no. 4, pp. 303–311, 2011 (cit. on pp. 4, 23).

[31] Q. Duan, C. Flynn, M. Niepel, M. Hafner, J. L. Muhlich, N. F. Fernandez, A. D. Rouillard, C. M. Tan, E. Y. Chen, T. R. Golub, *et al.*, "Lincs canvas browser: Interactive web app to query, browse and interrogate lincs l1000 gene expression signatures," *Nucleic acids research*, vol. 42, no. W1, W449–W460, 2014 (cit. on pp. 4, 23).

[32] R. Hodos, P. Zhang, H.-C. Lee, Q. Duan, Z. Wang, N. R. Clark, A. Maayan, F. Wang, B. Kidd, J. Hu, *et al.*, "Cell-specific prediction and application of drug-induced gene expression profiles," in *Pacific Symposium on Biocomputing*, World Scientific, vol. 23, 2017 (cit. on p. 4).

[33] I. Perros, F. Wang, P. Zhang, P. Walker, R. Vuduc, J. Pathak, and J. Sun, "Polyadic regression and its application to chemogenomics," in *Proceedings of the 2017 SIAM International Conference on Data Mining*, SIAM, 2017, pp. 72–80 (cit. on p. 4).

[34] *Electronic health records-based phenotyping*, `https://sites.duke.edu/rethinkingclinicaltrials/ehr-phenotyping/`, Accessed: 2018-10-09 (cit. on p. 5).

[35] E. R. Dubberke, H. A. Nyazee, D. S. Yokoe, J. Mayer, K. B. Stevenson, J. E. Mangino, Y. M. Khan, V. J. Fraser, *et al.*, "Implementing automated surveillance for tracking clostridium difficile infection at multiple healthcare facilities," *Infection Control & Hospital Epidemiology*, vol. 33, no. 3, pp. 305–308, 2012 (cit. on p. 5).

[36] W.-Q. Wei and J. C. Denny, "Extracting research-quality phenotypes from electronic health records to support precision medicine," *Genome medicine*, vol. 7, no. 1, p. 41, 2015 (cit. on p. 5).

[37] J. R. Robinson, W.-Q. Wei, D. M. Roden, and J. C. Denny, "Defining phenotypes from clinical data to drive genomic research," 2018 (cit. on pp. 5, 89, 90).

[38] P. Yadav, M. Steinbach, V. Kumar, and G. Simon, "Mining electronic health records (ehrs): A survey," *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, p. 85, 2018 (cit. on pp. 5, 6, 35, 89, 107, 111).

[39] E. Campo, S. H. Swerdlow, N. L. Harris, S. Pileri, H. Stein, and E. S. Jaffe, "The 2008 who classification of lymphoid neoplasms and beyond: Evolving concepts and practical applications," *Blood*, blood–2011, 2011 (cit. on p. 5).

[40] J Henry, Y. Pylypchuk, T. Searcy, and V. Patel, "Adoption of electronic health record systems among us non-federal acute care hospitals: 2008-2015," *ONC Data Brief*, vol. 35, pp. 1–9, 2016 (cit. on p. 6).

[41] B. S. Glicksberg, K. W. Johnson, and J. T. Dudley, "The next generation of precision medicine: Observational studies, electronic health records, biobanks and continuous monitoring," *Human molecular genetics*, vol. 27, no. R1, R56–R62, 2018 (cit. on p. 6).

[42] K. McLintock, A. M. Russell, S. L. Alderson, R. West, A. House, K. Westerman, and R. Foy, "The effects of financial incentives for case finding for depression in patients with diabetes and coronary heart disease: Interrupted time series analysis," *BMJ open*, vol. 4, no. 8, e005178, 2014 (cit. on p. 6).

[43] V. N. Slee, "The international classification of diseases: Ninth revision (icd-9)," *Annals of internal medicine*, vol. 88, no. 3, pp. 424–426, 1978 (cit. on pp. 6, 48, 78, 116).

[44] G. Hripcsak, D. J. Albers, and A. Perotte, "Parameterizing time in electronic health record studies," *Journal of the American Medical Informatics Association*, vol. 22, no. 4, pp. 794–804, 2015 (cit. on p. 6).

[45] R. L. Richesson, J. Sun, J. Pathak, A. N. Kho, and J. C. Denny, "Clinical phenotyping in selected national networks: Demonstrating the need for high-throughput, portable, and computational methods," *Artificial Intelligence in Medicine*, vol. 71, pp. 57–61, 2016 (cit. on pp. 6, 7, 82).

[46] S. Joshi, S. Gunasekar, D. Sontag, and J. Ghosh, "Identifiable phenotyping using constrained non-negative matrix factorization," *ArXiv preprint arXiv:1608.00704*, 2016 (cit. on pp. 6, 93).

[47] S. Yu, A. Chakrabortty, K. P. Liao, T. Cai, A. N. Ananthakrishnan, V. S. Gainer, S. E. Churchill, P. Szolovits, S. N. Murphy, I. S. Kohane, *et al.*, "Surrogate-assisted feature extraction for high-throughput phenotyping," *Journal of the American Medical Informatics Association*, vol. 24, no. e1, e143–e149, 2016 (cit. on p. 6).

[48] I. Perros, E. E. Papalexakis, H. Park, R. Vuduc, X. Yan, C. Defilippi, W. F. Stewart, and J. Sun, "Sustain: Scalable unsupervised scoring for tensors and its application to phenotyping," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '18, London, United Kingdom: ACM, 2018, pp. 2080–2089, ISBN: 978-1-4503-5552-0 (cit. on pp. 7, 8, 93, 111, 114, 121–123).

[49] Y. Wang, R. Chen, J. Ghosh, J. C. Denny, A. Kho, Y. Chen, B. A. Malin, and J. Sun, "Rubik: Knowledge guided tensor factorization and completion for health

data analytics," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '15, New York, NY, USA: ACM, 2015, pp. 1265–1274 (cit. on pp. 7, 35, 58).

[50] S. Joshi, S. Gunasekar, D. Sontag, and G. Joydeep, "Identifiable phenotyping using constrained Non-Negative matrix factorization," in *Machine Learning for Healthcare Conference*, Dec. 2016, pp. 17–41 (cit. on pp. 7, 35, 58).

[51] I. Perros, E. E. Papalexakis, F. Wang, R. Vuduc, E. Searles, M. Thompson, and J. Sun, "Spartan: Scalable parafac2 for large & sparse data," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2017, pp. 375–384 (cit. on pp. 7–9, 35, 58, 91, 97, 98, 101, 107, 111, 114, 121, 122, 134, 139–141).

[52] A. Rusanov, N. G. Weiskopf, S. Wang, and C. Weng, "Hidden in plain sight: Bias towards sick patients when sampling patients with sufficient electronic health record data for research," *BMC medical informatics and decision making*, vol. 14, p. 51, Jun. 2014 (cit. on p. 7).

[53] G. Hripcsak, D. J. Albers, and A. Perotte, "Parameterizing time in electronic health record studies," *Journal of the American Medical Informatics Association: JAMIA*, vol. 22, no. 4, pp. 794–804, Jul. 2015 (cit. on pp. 7, 90).

[54] S. C. Bagley and R. B. Altman, "Computing disease incidence, prevalence and comorbidity from electronic medical records," *Journal of biomedical informatics*, vol. 63, pp. 108–111, Oct. 2016 (cit. on p. 7).

[55] B. A. Goldstein, N. A. Bhavsar, M. Phelan, and M. J. Pencina, "Controlling for informed presence bias due to the number of health encounters in an electronic health record," *American journal of epidemiology*, vol. 184, no. 11, pp. 847–855, Dec. 2016 (cit. on p. 7).

[56] K. Zheng, J. Gao, K. Y. Ngiam, B. C. Ooi, and W. L. J. Yip, "Resolving the bias in electronic medical records," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '17, New York, NY, USA: ACM, 2017, pp. 2171–2180 (cit. on p. 7).

[57] J. B. Leader, S. A. Pendergrass, A. Verma, D. J. Carey, D. N. Hartzel, M. D. Ritchie, and H. L. Kirchner, "Contrasting association results between existing phewas phenotype definition methods and five validated electronic phenotypes," in *AMIA Annual Symposium Proceedings*, American Medical Informatics Association, vol. 2015, 2015, p. 824 (cit. on pp. 8, 36).

[58] R. A. Harshman, "Parafac2: Mathematical and technical notes," *UCLA Working Papers in Phonetics*, vol. 22, pp. 30–44, 1972b (cit. on pp. 8, 9, 63, 67, 68, 90, 91, 95, 97, 112, 117, 120, 129, 134, 137, 140).

[59] H. A. Kiers, J. M. Ten Berge, and R. Bro, "Parafac2-part i. a direct fitting algorithm for the parafac2 model," *Journal of Chemometrics*, vol. 13, no. 3-4, pp. 275–294, 1999 (cit. on pp. 8, 63, 67–70, 79, 91, 98, 112, 120, 121, 123, 134, 140).

[60] H. A. L. Kiers, J. M. F. Ten Berge, and R. Bro, "Parafac2-part i. a direct fitting algorithm for the parafac2 model," *Journal of chemometrics*, vol. 13, no. 3-4, 275294, 1999 (cit. on pp. 8, 97).

[61] I. Perros, E. E. Papalexakis, R. Vuduc, E. Searles, and J. Sun, "Temporal phenotyping of medically complex children via parafac2 tensor factorization," *Journal of biomedical informatics*, p. 103 125, 2019 (cit. on pp. 9, 111, 112, 114, 129, 140).

[62] T. Hofmann, J. Puzicha, and M. I. Jordan, "Learning from dyadic data," *Advances in neural information processing systems*, pp. 466–472, 1999 (cit. on p. 12).

[63] K. A. O'Connor and B. L. Roth, "Finding new tricks for old drugs: An efficient route for public-sector drug discovery," *Nature reviews Drug discovery*, vol. 4, no. 12, pp. 1005–1014, 2005 (cit. on p. 13).

[64] M. Sharma, J. Zhou, J. Hu, and G. Karypis, "Feature-based factorized bilinear similarity model for cold-start top-n item recommendation," in *SDM*, SIAM, vol. 15, 2015, pp. 190–198 (cit. on p. 13).

[65] B. Jin, H. Yang, C. Xiao, P. Zhang, X. Wei, and F. Wang, "Multitask dyadic prediction and its application in prediction of adverse drug-drug interaction," in *AAAI 2017 (to appear)* (cit. on pp. 14, 30).

[66] P. Jain and I. S. Dhillon, "Provable inductive matrix completion," *ArXiv preprint arXiv:1306.0626*, 2013 (cit. on pp. 14, 30).

[67] N. Natarajan and I. S. Dhillon, "Inductive matrix completion for predicting gene–disease associations," *Bioinformatics*, vol. 30, no. 12, pp. i60–i68, 2014 (cit. on pp. 14, 20, 30).

[68] D. B. Allison, G. P. Page, T. M. Beasley, and J. W. Edwards, *DNA microarrays and related genomics techniques: Design, analysis, and interpretation of experiments*. CRC Press, 2005 (cit. on pp. 15, 25).

[69] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 26, no. 8, pp. 1819–1837, 2014 (cit. on p. 17).

[70] T. Evgeniou and M. Pontil, "Regularized multi–task learning," in *KDD*, ACM, 2004, pp. 109–117 (cit. on p. 17).

[71] S. Rendle, "Factorization machines," in *ICDM*, IEEE, 2010, pp. 995–1000 (cit. on pp. 19, 25).

[72] S. Ji and J. Ye, "An accelerated gradient method for trace norm minimization," in *ICML*, ACM, 2009, pp. 457–464 (cit. on p. 19).

[73] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966 (cit. on pp. 20, 95).

[74] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," arXiv:1606.04838, Tech. Rep., 2016 (cit. on p. 21).

[75] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in optimization*, vol. 1, no. 3, pp. 123–231, 2013 (cit. on p. 22).

[76] J.-J. Moreau, "Fonctions convexes duales et points proximaux dans un espace hilbertien," *CR Acad. Sci. Paris Sér. A Math*, vol. 255, pp. 2897–2899, 1962 (cit. on p. 22).

[77] G. Wei, D. Twomey, *et al.*, "Gene expression-based chemical genomics identifies rapamycin as a modulator of mcl1 and glucocorticoid resistance," *Cancer cell*, vol. 10, no. 4, pp. 331–342, 2006 (cit. on p. 23).

[78] *Pubchem substructure fingerprints*, `https://pubchem.ncbi.nlm.nih.gov/`, Accessed: 2016-09-19 (cit. on p. 24).

[79] G. Yu, F. Li, *et al.*, "Gosemsim: An r package for measuring semantic similarity among go terms and gene products," *Bioinformatics*, vol. 26, no. 7, pp. 976–978, 2010 (cit. on p. 24).

[80] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, vol. 33, no. 1, pp. 1–22, 2010 (cit. on p. 25).

[81] S. Rendle, "Factorization machines with libfm," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 3, no. 3, p. 57, 2012 (cit. on p. 25).

[82] B. Cao, H. Zhou, G. Li, and P. S. Yu, "Multi-view machines," in *WSDM 2016*, pp. 427–436 (cit. on pp. 25, 28, 31).

[83] B. Zhao, G. Li, H. Zhou, and S. Li, *Zen*, `https://github.com/cloudml/zen`, 2016 (cit. on p. 25).

[84] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011 (cit. on p. 25).

[85] B. O'Donoghue, *Apg*, `https://github.com/bodono/apg`, 2016 (cit. on p. 26).

[86] A. K. Menon and C. Elkan, "A log-linear model with latent features for dyadic prediction," in *ICDM*, IEEE, 2010, pp. 364–373 (cit. on p. 30).

[87] J. Xu, K. Lin, P. Tan, and J. Zhou, "Synergies that matter: Efficient interaction selection via sparse factorization machine," in *SDM 2016*, pp. 108–116 (cit. on p. 30).

[88] Y. Huang, W. Wang, and L. Wang, "Conditional high-order boltzmann machine: A supervised learning model for relation learning," in *ICCV*, 2015, pp. 4265–4273 (cit. on p. 31).

[89] R. A. Harshman, "Foundations of the parafac procedure: Models and conditions for an "explanatory" multi-modal factor analysis," *UCLA Working Papers in Phonetics*, 1970 (cit. on pp. 31, 44, 62, 65, 66, 139).

[90] C. Xu, D. Tao, and C. Xu, "A survey on multi-view learning," *ArXiv preprint arXiv:1304.5634*, 2013 (cit. on p. 31).

[91] S. M. Kakade and D. P. Foster, "Multi-view regression via canonical correlation analysis," in *International Conference on Computational Learning Theory*, Springer, 2007, pp. 82–96 (cit. on p. 31).

[92] M. Gönen and E. Alpaydın, "Multiple kernel learning algorithms," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2211–2268, 2011 (cit. on p. 31).

[93] M. T. Bahadori, Q. R. Yu, and Y. Liu, "Fast multivariate spatio-temporal analysis via low rank tensor learning," in *NIPS*, 2014, pp. 3491–3499 (cit. on p. 31).

[94] B. Dong, M. M. Lin, and H. Park, "Integer matrix approximation and data mining," *Journal of scientific computing*, pp. 1–27, Sep. 2017 (cit. on pp. 34, 36, 50, 58).

[95] R. L. Richesson, J. Sun, J. Pathak, A. N. Kho, and J. C. Denny, "Clinical phenotyping in selected national networks: Demonstrating the need for high-throughput, portable, and computational methods," *Artificial intelligence in medicine*, vol. 71, pp. 57–61, Jul. 2016 (cit. on p. 35).

[96] X w. Chang and Q Han, "Solving Box-Constrained integer least squares problems," *IEEE Transactions on Wireless Communications*, vol. 7, no. 1, pp. 277–287, Jan. 2008 (cit. on pp. 36, 50).

[97] S. Breen and X.-W. Chang, "Column reordering for Box-Constrained integer least squares problems," Apr. 2012. arXiv: `1204.1407 [cs.IT]` (cit. on pp. 36, 50).

[98] J. C. Denny, M. D. Ritchie, M. A. Basford, J. M. Pulley, L. Bastarache, K. Brown-Gentry, D. Wang, D. R. Masys, D. M. Roden, and D. C. Crawford, "Phewas: Demonstrating the feasibility of a phenome-wide scan to discover gene–disease associations," *Bioinformatics*, vol. 26, no. 9, pp. 1205–1210, 2010 (cit. on p. 36).

[99] B. Ustun and C. Rudin, "Optimized risk scores," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2017, pp. 1125–1134 (cit. on p. 37).

[100] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004 (cit. on p. 38).

[101] A. Smilde, R. Bro, and P. Geladi, *Multi-way analysis: Applications in the chemical sciences*. John Wiley & Sons, 2005 (cit. on p. 38).

[102] B. W. Bader and T. G. Kolda, "Efficient matlab computations with sparse and factored tensors," *SIAM Journal on Scientific Computing*, vol. 30, no. 1, pp. 205–231, 2007 (cit. on pp. 39, 45, 46, 62, 67, 70, 71).

[103] R. Kannan, G. Ballard, and H. Park, "Mpi-faun: An mpi-based framework for alternating-updating nonnegative matrix factorization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 3, pp. 544–558, 2018 (cit. on p. 40).

[104] T. G. Kolda and D. P. O'Leary, "A semidiscrete matrix decomposition for latent semantic indexing information retrieval," *ACM Transactions on Information and System Security*, vol. 16, no. 4, pp. 322–346, Oct. 1998 (cit. on pp. 41, 58).

[105] N. Gillis *et al.*, "Nonnegative matrix factorization: Complexity, algorithms and applications," *Unpublished doctoral dissertation, Université catholique de Louvain. Louvain-La-Neuve: CORE*, 2011 (cit. on p. 41).

[106] R. Bro and N. D. Sidiropoulos, "Least squares algorithms under unimodality and non-negativity constraints," *Journal of Chemometrics*, vol. 12, no. 4, pp. 223–247, 1998 (cit. on pp. 42, 46).

[107] R. Takapoui, N. Moehle, S. Boyd, and A. Bemporad, "A simple effective heuristic for embedded mixed-integer quadratic programming," *International Journal of Control*, pp. 1–11, 2017 (cit. on p. 42).

[108] J. D. Carroll and J.-J. Chang, "Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970 (cit. on pp. 44, 62, 65, 66).

[109] G. H. Golub and C. F. Van Loan, *Matrix Computations*. JHU Press, 2013, vol. 3 (cit. on pp. 45, 69, 121).

[110] E. Choi, A. Schuetz, W. F. Stewart, and J. Sun, "Using recurrent neural network models for early detection of heart failure onset," *Journal of the American Medical Informatics Association*, vol. 24, no. 2, pp. 361–370, 2016 (cit. on pp. 48, 113).

[111] *Clinical classifications software (ccs) for icd-9-cm*, `https://www.hcup-us.ahrq.gov/toolssoftware/ccs/ccs.jsp`, Accessed: 2017-02-11, 2017 (cit. on pp. 48, 78).

[112] G. Nahler, "Anatomical therapeutic chemical classification system (atc)," in *Dictionary of Pharmaceutical Medicine*. Vienna: Springer Vienna, 2009, pp. 8–8, ISBN: 978-3-211-89836-9 (cit. on pp. 48, 116, 122).

[113] J. Kim and H. Park, "Fast nonnegative matrix factorization: An active-set-like method and comparisons," *SIAM Journal on Scientific Computing*, vol. 33, no. 6, pp. 3261–3281, 2011 (cit. on p. 49).

[114] B. W. Bader, T. G. Kolda, *et al.*, *Matlab tensor toolbox version 2.6*, Available online, 2015 (cit. on pp. 49, 52, 62, 67, 79, 122).

[115] J. Brewer, "Kronecker products and matrix calculus in system theory," *IEEE Transactions on circuits and systems*, vol. 25, no. 9, pp. 772–781, 1978 (cit. on p. 50).

[116] X.-W. Chang and T. Zhou, "Miles: Matlab package for solving mixed integer least squares problems," *GPS Solutions*, vol. 11, no. 4, pp. 289–294, 2007, Last updated: June 2016 (cit. on p. 52).

[117] S. Wu, A. Joseph, A. S. Hammonds, S. E. Celniker, B. Yu, and E. Frise, "Stability-driven nonnegative matrix factorization to interpret spatial gene expression and build local gene networks," *Proceedings of the National Academy of Sciences*, vol. 113, no. 16, pp. 4290–4295, 2016 (cit. on pp. 55, 56, 121).

[118] D O'Leary and S Peleg, "Digital image compression by outer product expansion," *IEEE Transactions on Communications*, vol. 31, no. 3, pp. 441–444, Mar. 1983 (cit. on p. 58).

[119] M. Koyutürk, A. Grama, and N. Ramakrishnan, "Nonorthogonal decomposition of binary matrices for bounded-error data compression and analysis," *ACM Transac-*

*tions on Mathematical Software (TOMS)*, vol. 32, no. 1, pp. 33–69, 2006 (cit. on p. 58).

[120] M. Koyuturk, A. Grama, and N. Ramakrishnan, "Compression, clustering, and pattern discovery in very high-dimensional discrete-attribute data sets," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 447–461, 2005 (cit. on p. 58).

[121] Z Zhang, T Li, C Ding, and X Zhang, "Binary matrix factorization with applications," in *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, Oct. 2007, pp. 391–400 (cit. on p. 58).

[122] B.-H. Shen, S. Ji, and J. Ye, "Mining discrete patterns via binary matrix factorization," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '09, New York, NY, USA: ACM, 2009, pp. 757–766 (cit. on p. 58).

[123] P Miettinen, T Mielikäinen, A Gionis, G Das, and H Mannila, "The discrete basis problem," *IEEE transactions on knowledge and data engineering*, vol. 20, no. 10, pp. 1348–1362, Oct. 2008 (cit. on p. 58).

[124] P. Miettinen, "Boolean tensor factorizations," in *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, IEEE, 2011, pp. 447–456 (cit. on p. 58).

[125] D. Lian, R. Liu, Y. Ge, K. Zheng, X. Xie, and L. Cao, "Discrete content-aware matrix factorization," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2017, pp. 325–334 (cit. on p. 58).

[126] J Henderson, J. C. Ho, A. N. Kho, J. C. Denny, B. A. Malin, J Sun, and J Ghosh, "Granite: Diversified, sparse tensor factorization for electronic health Record-Based phenotyping," in *2017 IEEE International Conference on Healthcare Informatics (ICHI)*, Aug. 2017, pp. 214–223 (cit. on pp. 58, 114).

[127] A. Cichocki and A.-H. Phan, "Fast local algorithms for large scale nonnegative matrix and tensor factorizations," *IEICE transactions on fundamentals of electronics, communications and computer sciences*, vol. 92, no. 3, pp. 708–721, 2009 (cit. on p. 58).

[128] N.-D. Ho, "Nonnegative matrix factorization algorithms and applications," PhD thesis, PhD thesis, Université catholique de Louvain, 2008 (cit. on p. 58).

[129] J. B. Kruskal, "Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics," *Linear algebra and its applications*, vol. 18, no. 2, pp. 95–138, 1977 (cit. on pp. 62, 66).

[130] N. D. Sidiropoulos and R. Bro, "On the uniqueness of multilinear decomposition of n-way arrays," *Journal of chemometrics*, vol. 14, no. 3, pp. 229–239, 2000 (cit. on pp. 62, 66).

[131] E. E. Papalexakis, C. Faloutsos, and N. D. Sidiropoulos, "Tensors for data mining and data fusion: Models, applications, and scalable algorithms," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8, no. 2, p. 16, 2016 (cit. on pp. 62, 66, 97, 140).

[132] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *ArXiv preprint arXiv:1607.01668*, 2016 (cit. on pp. 62, 66).

[133] E. C. Chi and T. G. Kolda, "On tensors, sparsity, and nonnegative factorizations," *SIAM Journal on Matrix Analysis and Applications*, vol. 33, no. 4, pp. 1272–1299, 2012 (cit. on p. 62).

[134] J. C. Ho, J. Ghosh, and J. Sun, "Marble: High-throughput phenotyping from electronic health records via sparse nonnegative tensor factorization," in *KDD*, ACM, 2014, pp. 115–124 (cit. on pp. 62, 93, 111, 114).

[135] Y. Wang, R. Chen, J. Ghosh, J. C. Denny, A. Kho, Y. Chen, B. A. Malin, and J. Sun, "Rubik: Knowledge guided tensor factorization and completion for health data analytics," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2015, pp. 1265–1274 (cit. on pp. 62, 93, 114).

[136] J. Sun, C. E. Tsourakakis, E. Hoke, C. Faloutsos, and T. Eliassi-Rad, "Two heads better than one: Pattern discovery in time-evolving multi-aspect data," *Data Mining and Knowledge Discovery*, vol. 17, no. 1, pp. 111–128, 2008 (cit. on p. 62).

[137] J. Zhou, F. Wang, J. Hu, and J. Ye, "From micro to macro: Data driven phenotyping by densification of longitudinal electronic medical records," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2014, pp. 135–144 (cit. on p. 62).

[138] F. Wang, N. Lee, J. Hu, J. Sun, S. Ebadollahi, and A. F. Laine, "A framework for mining signatures from event sequences and its applications in healthcare data," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 2, pp. 272–285, 2013 (cit. on pp. 62, 93).

[139] F. Wang, J. Zhou, and J. Hu, "Densitytransfer: A data driven approach for imputing electronic health records," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*, IEEE, 2014, pp. 2763–2768 (cit. on pp. 62, 93, 114).

[140] R. A. Harshman and M. E. Lundy, "Uniqueness proof for a family of models sharing features of tucker's three-mode factor analysis and parafac/candecomp," *Psychometrika*, vol. 61, no. 1, pp. 133–154, 1996 (cit. on pp. 63, 68).

[141] J. M. ten Berge and H. A. Kiers, "Some uniqueness results for parafac2," *Psychometrika*, vol. 61, no. 1, pp. 123–132, 1996 (cit. on pp. 63, 68).

[142] A. Stegeman and T. T. Lam, "Multi-set factor analysis by means of parafac2," *British Journal of Mathematical and Statistical Psychology*, 2015 (cit. on pp. 63, 68).

[143] R. Bro, "Parafac. tutorial and applications," *Chemometrics and intelligent laboratory systems*, vol. 38, no. 2, pp. 149–171, 1997 (cit. on pp. 63, 66, 86).

[144] P. A. Chew, B. W. Bader, T. G. Kolda, and A. Abdelali, "Cross-language information retrieval using parafac2," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2007, pp. 143–152 (cit. on pp. 63, 66, 68, 79).

[145] R Bro, "Multi-way analysis in the food industry," 1998 (cit. on pp. 65, 69, 97, 98, 101, 140).

[146] R. Bro and S. De Jong, "A fast non-negativity-constrained least squares algorithm," *Journal of chemometrics*, vol. 11, no. 5, pp. 393–401, 1997 (cit. on pp. 66, 79).

[147] U Kang, E. Papalexakis, A. Harpale, and C. Faloutsos, "Gigatensor: Scaling tensor analysis up by 100 times-algorithms and discoveries," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2012, pp. 316–324 (cit. on pp. 67, 86).

[148] J. H. Choi and S Vishwanathan, "Dfacto: Distributed factorization of tensors," in *Advances in Neural Information Processing Systems*, 2014, pp. 1296–1304 (cit. on pp. 67, 76).

[149] E. E. Papalexakis, C. Faloutsos, and N. D. Sidiropoulos, "Parcube: Sparse parallelizable candecomp-parafac tensor decomposition," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 10, no. 1, p. 3, 2015 (cit. on p. 67).

[150] S. Smith, N. Ravindran, N. D. Sidiropoulos, and G. Karypis, "Splatt: Efficient and parallel sparse tensor-matrix multiplication," in *Parallel and Distributed Processing Symposium (IPDPS), 2015 IEEE International*, IEEE, 2015, pp. 61–70 (cit. on pp. 67, 70).

[151] D. Cheng, R. Peng, I. Perros, and Y. Liu, "Spals: Fast alternating least squares via implicit leverage scores sampling," in *Advances In Neural Information Processing Systems*, 2016, pp. 721–729 (cit. on p. 67).

[152] E. Acar and B. Yener, "Unsupervised multiway data analysis: A literature survey," *IEEE transactions on knowledge and data engineering*, vol. 21, no. 1, pp. 6–20, 2009 (cit. on p. 68).

[153] N. E. Helwig, "The special sign indeterminacy of the direct-fitting parafac2 model: Some implications, cautions, and recommendations for simultaneous component analysis," *Psychometrika*, vol. 78, no. 4, pp. 725–739, 2013 (cit. on p. 68).

[154] R. Bro, C. A. Andersson, and H. A. Kiers, "Parafac2-part ii. modeling chromatographic data with retention time shifts," *Journal of Chemometrics*, vol. 13, no. 3-4, pp. 295–309, 1999 (cit. on pp. 70, 101, 140).

[155] L. N. Trefethen and D. Bau III, *Numerical linear algebra*, 1997 (cit. on p. 70).

[156] N. Lathia, S. Hailes, L. Capra, and X. Amatriain, "Temporal diversity in recommender systems," in *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, ACM, 2010, pp. 210–217 (cit. on p. 78).

[157] C. Andersson and R. Bro, *The n-way toolbox for matlab*, Available online, 2000 (cit. on pp. 79, 101, 122).

[158] K. H. Burns, P. H. Casey, R. E. Lyle, T. M. Bird, J. J. Fussell, and J. M. Robbins, "Increasing prevalence of medically complex children in us hospitals," *Pediatrics*, vol. 126, no. 4, 638646, Oct. 2010 (cit. on pp. 88, 89).

[159] P. W. Newacheck, B. Strickland, J. P. Shonkoff, J. M. Perrin, M. McPherson, M. McManus, C. Lauver, H. Fox, and P. Arango, "An epidemiologic profile of children with special health care needs," *Pediatrics*, vol. 102, no. 1, pp. 117–123, 1998 (cit. on p. 88).

[160] P. B. Edwin F. Simpser, *Children and Young Adults with Medical Complexity: Serving an Emerging Population*. Jan. 2016 (cit. on p. 88).

[161] T. D. Simon, J. Berry, C. Feudtner, B. L. Stone, X. Sheng, S. L. Bratton, J. M. Dean, and R. Srivastava, "Children with complex chronic conditions in inpatient hospital settings in the united states," *Pediatrics*, vol. 126, no. 4, 647655, Oct. 2010 (cit. on p. 89).

[162] C. M. Clancy and E. M. Andresen, "Meeting the health care needs of persons with disabilities," *The Milbank quarterly*, vol. 80, no. 2, 381391, 2002 (cit. on p. 89).

[163] C. H. Association, "Optimizing health care for children with medical complexity," (cit. on p. 89).

[164] C. Feudtner, J. E. Levin, R. Srivastava, D. M. Goodman, A. D. Slonim, V. Sharma, S. S. Shah, S. Pati, C. Fargason Jr, and M. Hall, "How well can hospital readmission be predicted in a cohort of hospitalized children? a retrospective, multicenter study," *Pediatrics*, vol. 123, no. 1, 286293, Jan. 2009 (cit. on p. 89).

[165] O. Gottesman, H. Kuivaniemi, G. Tromp, W. A. Faucett, R. Li, T. A. Manolio, S. C. Sanderson, J. Kannry, R. Zinberg, M. A. Basford, *et al.*, "The electronic medical records and genomics (emerge) network: Past, present, and future," *Genetics in Medicine*, vol. 15, no. 10, p. 761, 2013 (cit. on pp. 89, 111).

[166] J. C. Kirby, P. Speltz, L. V. Rasmussen, M. Basford, O. Gottesman, P. L. Peissig, J. A. Pacheco, G. Tromp, J. Pathak, D. S. Carrell, *et al.*, "Phekb: A catalog and workflow for creating electronic phenotype algorithms for transportability," *Journal of the American Medical Informatics Association*, vol. 23, no. 6, pp. 1046–1052, 2016 (cit. on pp. 89, 111).

[167] B. S. Glicksberg, R. Miotto, K. W. Johnson, K. Shameer, L. Li, R. Chen, and J. T. Dudley, "Automated disease cohort selection using word embeddings from electronic health records," in *Pac Symp Biocomput*, World Scientific, vol. 23, 2018, pp. 145–56 (cit. on pp. 89, 94, 111, 114).

[168] G. Hripcsak and D. J. Albers, "Next-generation phenotyping of electronic health records," *Journal of the American Medical Informatics Association: JAMIA*, vol. 20, no. 1, pp. 117–121, Jan. 2013 (cit. on p. 90).

[169] Y Cheng, F Wang, P Zhang, and J Hu, "Risk prediction with electronic health records: A deep learning approach," in *Proceedings of the 2016 SIAM International Conference on Data Mining*, ser. Proceedings, Society for Industrial and Applied Mathematics, 2016, 432440 (cit. on p. 90).

[170] G. Hripcsak, "Physics of the medical record: Handling time in health record studies," ser. Lecture Notes in Computer Science, Springer, Cham, 2015, 36 (cit. on p. 90).

[171] R. Bro and H. A. L. Kiers, "A new efficient method for determining the number of components in parafac models," *Journal of chemometrics*, vol. 17, no. 5, 274286, 2003 (cit. on pp. 91, 99, 101, 102, 140).

[172] C. Shivade, P. Raghavan, E. Fosler-Lussier, P. J. Embi, N. Elhadad, S. B. Johnson, and A. M. Lai, "A review of approaches to identifying patient phenotype cohorts using electronic health records," *Journal of the American Medical Informatics Association*, vol. 21, no. 2, pp. 221–230, 2013 (cit. on p. 92).

[173] T. A. Lasko, J. C. Denny, and M. A. Levy, "Computational phenotype discovery using unsupervised feature learning over noisy, sparse, and irregular clinical data," *PloS one*, vol. 8, no. 6, e66341, 2013 (cit. on pp. 92, 114).

[174] D. C. Belgrave, R. Granell, A. Simpson, J. Guiver, C. Bishop, I. Buchan, A. J. Henderson, and A. Custovic, "Developmental profiles of eczema, wheeze, and rhinitis: Two population-based birth cohort studies," *PLoS medicine*, vol. 11, no. 10, e1001748, 2014 (cit. on pp. 92, 115).

[175] M. Ghassemi, M. A. F. Pimentel, T. Naumann, T. Brennan, D. A. Clifton, P. Szolovits, and M. Feng, "A multivariate timeseries modeling approach to severity of illness assessment and forecasting in icu with sparse, heterogeneous clinical data," *Proceedings of the AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, vol. 2015, 446453, Jan. 2015 (cit. on pp. 92, 114).

[176] R. Pivovarov, A. J. Perotte, E. Grave, J. Angiolillo, C. H. Wiggins, and N. Elhadad, "Learning probabilistic phenotypes from heterogeneous ehr data," *Journal of biomedical informatics*, vol. 58, pp. 156–165, 2015 (cit. on pp. 92, 115).

[177] J. Zhou, F. Wang, J. Hu, and J. Ye, "From micro to macro: Data driven phenotyping by densification of longitudinal electronic medical records," ser. KDD '14, New York, NY, USA: ACM, 2014, 135144 (cit. on pp. 93, 114).

[178] A. Schuler, V. Liu, J. Wan, A. Callahan, M. Udell, D. E. Stark, and N. H. Shah, "Discovering patient phenotypes using generalized low rank models," 2015 (cit. on pp. 93, 114).

[179] M. Ruffini, R. Gavald, and E. Limn, "Clustering patients with tensor decomposition," 2017 (cit. on pp. 93, 114).

[180] J. Henderson, B. A. Malin, J. C. Ho, and J. Ghosh, "Piveted-granite: computational phenotypes through constrained tensor factorization," *ArXiv e-prints*, Aug. 2018. arXiv: 1808.02602 (cit. on p. 93).

[181] S. Gunasekar, J. C. Ho, J. Ghosh, S. Kreml, A. N. Kho, J. C. Denny, B. A. Malin, and J. Sun, "Phenotyping using structured collective matrix factorization of multi–source ehr data," *ArXiv preprint arXiv:1609.04466*, 2016 (cit. on p. 93).

[182] K. B. Kshetri, "Modelling patient states in intensive care patients," PhD thesis, 2011 (cit. on pp. 93, 114).

[183] F. Doshi-Velez, Y. Ge, and I. Kohane, "Comorbidity clusters in autism spectrum disorders: An electronic health record time-series analysis," *Pediatrics*, vol. 133, no. 1, e54, 2014 (cit. on pp. 93, 114).

[184] P. Schulam, F. Wigley, and S. Saria, "Clustering longitudinal clinical marker trajectories from electronic health data: Applications to phenotyping and endotype discovery," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015 (cit. on pp. 93, 114).

[185] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119 (cit. on pp. 94, 114).

[186] E. Choi, M. T. Bahadori, E. Searles, C. Coffey, M. Thompson, J. Bost, J. Tejedor-Sojo, and J. Sun, "Multi-layer representation learning for medical concepts," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2016, pp. 1495–1504 (cit. on pp. 94, 114).

[187] A. Dagliati, V. Tibollo, G. Cogni, L. Chiovato, R. Bellazzi, and L. Sacchi, "Careflow mining techniques to explore type 2 diabetes evolution," *Journal of diabetes science and technology*, vol. 12, no. 2, pp. 251–259, 2018 (cit. on pp. 94, 115).

[188] J. L. Warner, A. Zollanvari, Q. Ding, P. Zhang, G. M. Snyder, and G. Alterovitz, "Temporal phenome analysis of a large electronic health record cohort enables identification of hospital-acquired complications," *Journal of the American Medical Informatics Association*, vol. 20, no. e2, e281–e287, 2013 (cit. on pp. 94, 115).

[189] R. Bro, "Multi-way analysis in the food industry - models, algorithms, and applications," Tech. Rep., 1998 (cit. on p. 96).

[190] M. H. Kamstrup-Nielsen, L. G. Johnsen, and R. Bro, "Core consistency diagnostic in parafac2," *Journal of Chemometrics*, vol. 27, no. 5, pp. 99–105, May 1, 2013 (cit. on pp. 97, 98, 140, 141).

[191] R. Bro and H. A. L. Kiers, "A new efficient method for determining the number of components in PARAFAC models," *Journal of chemometrics*, vol. 17, no. 5, pp. 274–286, May 2003 (cit. on p. 97).

[192] E. E. Papalexakis and C. Faloutsos, "Fast efficient and scalable core consistency diagnostic for the parafac decomposition for big sparse tensors," 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Apr. 2015, pp. 5441–5445 (cit. on p. 98).

[193] R. Bro, C. A. Andersson, and H. A. L. Kiers, "Parafac2part ii. modeling chromatographic data with retention time shifts," *Journal of Chemometrics*, vol. 13, no. 3-4, pp. 295–309, May 1, 1999 (cit. on p. 98).

[194] v. d. L. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008 (cit. on pp. 99, 106, 141, 144).

[195] *HCUP-US Tools and Software Page CCS-Services and Procedures* (cit. on pp. 100, 122).

[196] *CPT (Current Procedural Terminology) | American Medical Association* (cit. on p. 100).

[197] A. for Healthcare Research and Quality, *Clinical decision support*, https://www.ahrq.gov/professionals/prevention-chronic-care/decision/clinical/index.html (cit. on p. 107).

[198] R. Richesson and M. Smerek, *Rethinking clinical trials*, https://sites.duke.edu/rethinkingclinicaltrials/ehr-phenotyping (cit. on p. 107).

[199] Clinithink, *A paradigm shift in patient recruitment for clinical trials*, https://clinithink.com/white-paper-paradigm-shift-patient-recruitment/ (cit. on p. 108).

[200] A. Uijl, S. Koudstaal, K. Direk, S. Denaxas, R. H. H. Groenwold, A. Banerjee, A. W. Hoes, H. Hemingway, and F. W. Asselbergs, "Risk factors for incident heart failure in age- and sex-specific strata: A population-based cohort using linked electronic health records," *European Journal of Heart Failure*, Jan. 7, 2019, PMID: 30618162 (cit. on pp. 110, 113).

[201] J. G. Westphal, T. Bekfani, and P. C. Schulze, "Whats new in heart failure therapy 2018?" *Interactive CardioVascular and Thoracic Surgery*, vol. 27, no. 6, pp. 921–930, Dec. 1, 2018 (cit. on p. 110).

[202] *Clinical phenotypes in heart failure with preserved ejection fraction. - pubmed - ncbi*, [Online; accessed 2019-04-14] (cit. on p. 110).

[203] S. J. Shah, D. H. Katz, S. Selvaraj, M. A. Burke, C. W. Yancy, M. Gheorghiade, R. O. Bonow, C.-C. Huang, and R. C. Deo, "Phenomapping for novel classification of heart failure with preserved ejection fraction," *Circulation*, vol. 131, no. 3, pp. 269–279, Jan. 20, 2015, PMID: 25398313 PMCID: PMC4302027 (cit. on pp. 110, 113).

[204] A. Afshar, I. Perros, E. E. Papalexakis, E. Searles, J. Ho, and J. Sun, "Copa: Constrained parafac2 for sparse & large datasets," ser. CIKM '18, event-place: Torino, Italy, New York, NY, USA: ACM, 2018, 793802, ISBN: 978-1-4503-6014-2 (cit. on pp. 111, 114).

[205] N. Kenney, S. S. R., deFilippi Christopher, D. Sanjoy, and S. W. F., "Early detection of heart failure using electronic health records," *Circulation: Cardiovascular Quality and Outcomes*, vol. 9, no. 6, pp. 649–658, Nov. 1, 2016 (cit. on pp. 113, 122).

[206] R. Vijayakrishnan, S. R. Steinhubl, K. Ng, J. Sun, R. J. Byrd, Z. Daar, B. A. Williams, C. deFilippi, S. Ebadollahi, and W. F. Stewart, "Prevalence of heart failure signs and symptoms in a large primary care population identified through the use of text and data mining of the electronic health record," *Journal of Cardiac Failure*, vol. 20, no. 7, pp. 459–464, Jul. 1, 2014 (cit. on pp. 113, 116).

[207] D. Knorek, S. Steinhubl, C. deFilippi, K. Ng, R. Byrd, Z. Daar, and W. Stewart, "Loop diuretic use in the months and years preceding a heart failure diagnosis: A case-control study," *Journal of Patient-Centered Research and Reviews*, vol. 4, no. 3, p. 163, Aug. 10, 2017 (cit. on p. 113).

[208] K. Ng, W. Stewart, C. deFilippi, S. Dey, R. Byrd, S. Steinhubl, Z. Daar, H. Law, A. Pressmnan, and J. Hu, "Data-driven modeling of electronic health record data to predict prediagnostic heart failure in primary care," *Journal of Patient-Centered Research and Reviews*, vol. 3, no. 3, p. 200, Aug. 15, 2016 (cit. on p. 113).

[209] K. Yin, D. Qian, W. K. Cheung, B. C. M. Fung, and J. Poon, "Learning phenotypes and dynamic patient representations via rnn regularized collective non-negative tensor factorization," p. 8, (cit. on p. 114).

[210] I Perros, R Chen, R Vuduc, and J Sun, "Sparse hierarchical tucker factorization and its application to healthcare," Nov. 2015, 943948 (cit. on p. 114).

[211] J. Henderson, B. A. Malin, J. C. Ho, and J. Ghosh, "Piveted-granite: Computational phenotypes through constrained tensor factorization," *ArXiv:1808.02602 [cs, stat]*, Aug. 7, 2018, arXiv: 1808.02602 (cit. on p. 114).

[212] *Eventthread: Visual summarization and stage analysis of event sequence data. - pubmed - ncbi*, [Online; accessed 2019-04-01] (cit. on p. 114).

[213] B. K. Beaulieu-Jones, P. Orzechowski, and J. H. Moore, "Mapping patient trajectories using longitudinal extraction and deep learning in the mimic-iii critical care database," *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, vol. 23, pp. 123–132, 2018, PMID: 29218875 (cit. on p. 114).

[214] A. B. Jensen, P. L. Moseley, T. I. Oprea, S. G. Ellese, R. Eriksson, H. Schmock, P. B. Jensen, L. J. Jensen, and S. Brunak, "Temporal disease trajectories condensed from population-wide registry data covering 6.2 million patients," *Nature Communications*, vol. 5, no. 1, Dec. 2014, [Online; accessed 2019-03-31] (cit. on p. 115).

[215] *Tracing diagnosis trajectories over millions of inpatients reveal an unexpected association between schizophrenia and rhabdomyolysis | biorxiv*, [Online; accessed 2019-03-31] (cit. on p. 115).

[216] O. Ichikawa, B. S. Glicksberg, N. Genes, B. A. Kidd, L. Li, and J. T. Dudley, "Lyme disease patient trajectories learned from electronic medical data for stratification of disease risk and therapeutic response," *Scientific Reports*, vol. 9, no. 1, p. 4460, Mar. 14, 2019, PMID: 30872757 PMCID: PMC6418311 (cit. on p. 115).

[217] I. Perros, *SPARTan: Scalable PARAFAC2 for Large & Sparse Data*. Nov. 13, 2018, original-date: 2017-05-26T02:29:01Z (cit. on p. 122).

[218] ——, *SUSTain: Scalable Unsupervised Scoring for Tensors and its Application to Phenotyping*. Mar. 28, 2019, original-date: 2018-05-24T12:28:54Z (cit. on p. 122).

[219] P. Mazzocato, R. J. Holden, M. Brommels, H. Aronsson, U. Bckman, M. Elg, and J. Thor, "How does lean work in emergency care? a case study of a lean-inspired intervention at the astrid lindgren children's hospital, stockholm, sweden," *BMC Health Services Research*, vol. 12, p. 28, Feb. 1, 2012 (cit. on pp. 133, 146).

[220] B. G. Arndt, J. W. Beasley, M. D. Watkinson, J. L. Temte, W.-J. Tuan, C. A. Sinsky, and V. J. Gilchrist, "Tethered to the ehr: Primary care physician workload assessment using ehr event log data and time-motion observations," *Annals of Family Medicine*, vol. 15, no. 5, pp. 419–426, 2017, PMID: 28893811 PMCID: PMC5593724 (cit. on p. 133).

[221] G. Hripcsak, D. K. Vawdrey, M. R. Fred, and S. B. Bostwick, "Use of electronic clinical documentation: Time spent and team interactions," *Journal of the American Medical Informatics Association: JAMIA*, vol. 18, no. 2, pp. 112–117, Apr. 2011, PMID: 21292706 PMCID: PMC3116265 (cit. on p. 133).

[222] S. Read-Brown, M. R. Hribar, L. G. Reznick, L. H. Lombardi, M. Parikh, W. D. Chamberlain, S. T. Bailey, J. B. Wallace, T. R. Yackel, and M. F. Chiang, "Time requirements for electronic health record use in an academic ophthalmology center," *JAMA ophthalmology*, vol. 135, no. 11, pp. 1250–1257, Nov. 1, 2017, PMID: 29049512 PMCID: PMC5710390 (cit. on p. 133).

[223] A. G. Hirsch, J. B. Jones, V. R. Lerch, X. Tang, A. Berger, D. N. Clark, and W. F. Stewart, "The electronic health record audit file: The patient is waiting," *Journal of the American Medical Informatics Association: JAMIA*, vol. 24, no. e1, e28e34, 2017 (cit. on pp. 133, 134, 147).

[224] A. W. Ulwick, *Jobs to be done: Theory to practice*. Idea Bite Press, 2016 (cit. on pp. 134, 147).

[225] J. J. Cimino, "Improving the electronic health recordare clinicians getting what they wished for?" *JAMA*, vol. 309, no. 10, pp. 991–992, Mar. 13, 2013 (cit. on p. 146).

[226] T. Melton, "The benefits of lean manufacturing: What lean thinking has to offer the process industries," *Chemical engineering research and design*, vol. 83, no. 6, 662673, 2005 (cit. on p. 146).

[227] J. Moraros, M. Lemstra, and C. Nwankwo, "Lean interventions in healthcare: Do they actually work? a systematic literature review," *International Journal for Quality in Health Care*, vol. 28, no. 2, pp. 150–165, Apr. 1, 2016 (cit. on p. 146).

[228] C. M. Christensen, K. Dillon, T. Hall, and D. S. Duncan, *Competing against luck: The story of innovation and customer choice*. Harper Business, 2016 (cit. on p. 147).

[229] A. W. Ulwick, *What customers want using outcome-driven innovation to create breakthrough products and services*. New York: McGraw-Hill, 2005, OCLC: 704667772, ISBN: 978-0-07-140867-7 (cit. on p. 147).

[230] J. Adler-Milstein and R. S. Huckman, "The impact of electronic health record use on physician productivity," *The American Journal of Managed Care*, vol. 19, no. 10 Spec No, SP345–352, Nov. 2013, PMID: 24511889 (cit. on p. 147).

[231] M. Tai-Seale, C. W. Olson, J. Li, A. S. Chan, C. Morikawa, M. Durbin, W. Wang, and H. S. Luft, "Electronic health record logs indicate that physicians split time evenly between seeing patients and desktop medicine," *Health affairs*, vol. 36, no. 4, 655662, 2017 (cit. on p. 147).

[232] M. R. Hribar, S. Read-Brown, I. H. Goldstein, L. G. Reznick, L. Lombardi, M. Parikh, W. Chamberlain, and M. F. Chiang, "Secondary use of electronic health record data for clinical workflow analysis," *Journal of the American Medical Informatics Association: JAMIA*, Sep. 26, 2017, PMID: 29036581 (cit. on p. 147).

[233] *Development of a primary care physician task list to evaluate clinic visit workflow | bmj quality & safety*, [Online; accessed 2019-03-02] (cit. on p. 147).

[234] M. R. Cowie, S. D. Anker, J. G. Cleland, G. M. Felker, G. Filippatos, T. Jaarsma, P. Jourdain, E. Knight, B. Massie, P. Ponikowski, *et al.*, "Improving care for patients with acute heart failure: Before, during and after hospitalization," *ESC Heart Failure*, vol. 1, no. 2, pp. 110–145, 2014 (cit. on p. 151).

[235] X. Fu, K. Huang, W.-K. Ma, N. D. Sidiropoulos, and R. Bro, "Joint tensor factorization and outlying slab suppression with applications," *IEEE Transactions on Signal Processing*, vol. 63, no. 23, pp. 6315–6328, 2015 (cit. on p. 152).